

# 80C51-Based 8-Bit Microcontrollers



FUTURE ELECTRONICS INC.  
5935 Airport Road,  
Suite 200  
Mississauga, Ontario  
L4V 1W5  
TEL.: (416) 612-9200  
FAX: (416) 612-9185  
TOLL FREE 1-800-268-7948

## DATA HANDBOOK

Philips Semiconductors



# PHILIPS

B O O K | I C 2 0 | 1 9 9 4



## **QUALITY ASSURED**

Our quality system focuses on the continuing high quality of our components and the best possible service for our customers. We have a three-sided quality strategy: we apply a system of total quality control and assurance; we operate customer-oriented dynamic improvement programmes; and we promote a partnering relationship with our customers and suppliers.

## **PRODUCT SAFETY**

In striving for state-of-the-art perfection, we continuously improve components and processes with respect to environmental demands. Our components offer no hazard to the environment in normal use when operated or stored within the limits specified in the data sheet.

Some components unavoidably contain substances that, if exposed by accident or misuse, are potentially hazardous to health. Users of these components are informed of the danger by warning notices in the data sheets supporting the components. Where necessary the warning notices also indicate safety precautions to be taken and disposal instructions to be followed. Obviously users of these components, in general the set-making industry, assume responsibility towards the consumer with respect to safety matters and environmental demands.

All used or obsolete components should be disposed of according to the regulations applying at the disposal location. Depending on the location, electronic components are considered to be 'chemical', 'special' or sometimes 'industrial' waste. Disposal as domestic waste is usually not permitted.



# 80C51-Based 8-Bit Microcontrollers

## CONTENTS

	page
<b>SECTION 1 GENERAL INFORMATION</b>	5
<b>SECTION 2 80C51 TECHNICAL DESCRIPTION</b>	31
<b>SECTION 3 80C51 FAMILY DERIVATIVES</b>	133
<b>SECTION 4 INTER-INTEGRATED (I<sup>2</sup>C) CIRCUIT BUS</b>	1139
<b>SECTION 5 ACCESS.bus TECHNICAL OVERVIEW</b>	1167
<b>SECTION 6 CONTROL AREA NETWORK (CAN) BUS</b>	1183
<b>SECTION 7 OTHER MICROCONTROLLERS</b>	1255
<b>SECTION 8 DEVELOPMENT SUPPORT TOOLS</b>	1275
<b>SECTION 9 PACKAGE OUTLINES</b>	1353
<b>SECTION 10 NORTH AMERICAN SALES OFFICES, REPRESENTATIVES, AND DISTRIBUTORS</b>	1391
<b>APPENDIX A DATA HANDBOOK SYSTEM</b>	1394
<b>APPENDIX B PIN CONFIGURATIONS</b>	1396



# 80C51-Based 8-Bit Microcontrollers

## CONTENTS

Page

5

31

133

1130

1167

1183

1252

1275

1353

1391

1394

1396

Philips Semiconductors and Philips Electronics North America Corporation reserve the right to make changes, without notice, in the products, including circuits, standard cells, and/or software, described or contained herein in order to improve design and/or performance. Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified. Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

### LIFE SUPPORT APPLICATIONS

Philips Semiconductors and Philips Electronics North America Corporation Products are not designed for use in life support appliances, devices, or systems where malfunction of a Philips Semiconductors and Philips Electronics North America Corporation Product can reasonably be expected to result in a personal injury. Philips Semiconductors and Philips Electronics North America Corporation customers using or selling Philips Semiconductors and Philips Electronics North America Corporation Products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors and Philips Electronics North America Corporation for any damages resulting from such improper use or sale.

Philips Semiconductors and Philips Electronics North America Corporation register eligible circuits under the Semiconductor Chip Protection Act.

© Copyright Philips Electronics North America Corporation, 1994

All rights reserved.  
Printed in U.S.A.



## 80C51-Based 8-Bit Microcontrollers

### Microcontrollers from Philips Semiconductors

Philips Semiconductors supplies a wide range of microcontrollers based on mainstream architectures. By offering a large variety of product derivatives, Philips Semiconductors can meet a broad range of specific or unique application requirements. All of our microcontrollers are based on mainstream architectures to allow the user to take advantage of existing software and a vast array of third-party support.

Philips Semiconductors 8-bit microcontrollers are based on the popular 80C51 architecture. We offer most of the industry standard products in this architecture as well as a large selection of powerful derivative products. These derivatives offer a wide assortment of features, including: additional memory, A/D, PWM, additional timers, and many more. Many of the derivative microcontrollers have an I<sup>2</sup>C serial interface that allows them to be connected easily to over 100 other parts, increasing their capabilities even further. The I<sup>2</sup>C serial bus is covered in Section 4 of this book. Philips Semiconductors also offers the Controller Area Network (CAN) serial bus for automotive and industrial applications. This standard, developed by Bosch, offers high noise immunity and error correction for automotive and industrial environments. The CAN serial bus is covered in Section 6 of this book. The data sheets for Philips low power 80CL51 family of derivatives are included in Section 3. These devices operate over the wide voltage range of 1.8 - 6.0 volts and are ideal for portable and battery operations. This data book covers the 80C51 standard products and derivatives that Philips Semiconductors manufactures.

Philips Semiconductors 16-bit 90CXXX microcontroller family is based on the 68000 architecture. While these are called 16-bit microcontrollers, the 68000 CPU core architecture is 32-bit. This offers the user a great deal more processing power, when the need arises in a design to move from an 8-bit to a 16-bit microcontroller. Philips Semiconductors 16-bit microcontrollers are software compatible with existing 68000 code. As with our popular 8-bit microcontrollers, EPROM and OTP versions of our 16-bit products are available. The 16-bit microcontrollers are covered in a separate data book.

Philips Semiconductors offers uncompromising quality, service, and support with all of its microcontroller products. For a complete family and the best in microcontroller products, look to Philips Semiconductors.





# Section 1

## General Information

### 80C51-Based 8-Bit Microcontrollers

#### CONTENTS

Contents .....	7
Product Status .....	11
80C51 Microcontrollers family features guide .....	12
8051 microcontroller cross-reference guide .....	14
83CLXXX microcontroller family — low power 80CL51 family derivatives .....	15
80C51 microcontroller development system support .....	16
8-bit microcontroller demonstrations and evaluation boards .....	17
Microcontroller bulletin boards .....	18
CMOS and NMOS 8-bit microcontroller family .....	19
CMOS 16-bit microcontroller family .....	27
Ordering information .....	28





## CONTENTS

## CONTENTS

Preface .....	3
<b>Section 1 – General Information</b>	
Contents .....	7
Product Status .....	11
80C51 Microcontrollers family features guide .....	12
8051 microcontroller cross-reference guide .....	14
83CLXXX microcontroller family — low power 80C51 family derivatives .....	15
80C51 microcontroller development system support .....	16
8-bit microcontroller demonstrations and evaluation boards .....	17
Microcontroller bulletin boards .....	18
CMOS and NMOS 8-bit microcontroller family .....	19
CMOS 16-bit microcontroller family .....	27
Ordering information .....	28
<b>Section 2 – 80C51 Technical Description</b>	
80C51 architecture .....	33
80C51 hardware description .....	48
80C51 programmer's guide and instruction set .....	73
80C51 EPROM products .....	128
<b>Section 3 – 80C51 Family Derivatives</b>	
80C31/80C51/87C51 CMOS single-chip 8-bit microcontroller .....	135
83C51FA/87C51FA CMOS single-chip 8-bit microcontroller .....	153
83C51FB/87C51FB CMOS single-chip 8-bit microcontrollers .....	168
83C51FC/87C51FC CMOS single-chip 8-bit microcontrollers .....	194
80CL31/80CL51 Low-voltage single-chip 8-bit microcontrollers .....	220
83L51FA/87L51FA CMOS single-chip 3.0V 8-bit microcontroller .....	254
83L51FB/87L51FB CMOS single-chip 3.0V 8-bit microcontroller .....	269
80C32/80C52/87C52 CMOS single-chip 8-bit microcontrollers .....	284
83C055/87C055 Microcontroller for television and video (MTV) .....	304
P83CL168/167/268/267 Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function .....	323
80CL410/83CL410 Low voltage/low power single-chip 8-bit microcontroller with I <sup>2</sup> C .....	382
80C451/83C451/87C451 CMOS single-chip 8-bit microcontroller .....	405
83C504/87C504 CMOS single-chip 8-bit microcontroller .....	424
P83C524 8-bit microcontroller .....	440
87C524 CMOS single-chip 8-bit microcontroller .....	460
80C528/83C528 CMOS single-chip 8-bit microcontroller .....	478
87C528 CMOS single-chip 8-bit microcontroller .....	495
P8xC528 8-bit microcontroller with EMC and EEPROM .....	382
80C550/83C550/87C550 CMOS single-chip 8-bit microcontroller with A/D and watchdog timer .....	554
8XC552/562 overview .....	575
8XC552 OVERVIEW .....	575
83C562 OVERVIEW .....	575
Differences From the 80C51 .....	575
Program Memory .....	575
Data Memory .....	575
Special Function Registers .....	575
Timer T2 .....	576
Timer T3, The Watchdog Timer .....	582
Serial I/O .....	583
Reset Circuitry .....	616
Interrupts .....	617
I/O Port Structure .....	621
Port 1 Operation .....	621
Port 5 Operation .....	621
Pulse Width Modulated Outputs .....	621
Analog-to-Digital Converter .....	621
Power Reduction Modes .....	627
Memory Organization .....	629

## CONTENTS

80C552/83C552	Single-chip 8-bit microcontroller with 10-bit A/D, capture/compare timer, high-speed outputs, PWM	634
87C552	Single-chip 8-bit microcontroller with 10-bit A/D, capture/compare timer, high-speed outputs, PWM	654
80CE558/83CE558/89CE558	Single-chip 8-bit microcontroller	674
80C562/83C562	Single-chip 8-bit microcontroller with 8-bit A/D, capture/compare timer, high-speed outputs, PWM	727
80C575/83C575/87C575	CMOS single-chip 8-bit microcontroller	740
83C576/87C576	CMOS single-chip 8-bit microcontroller	774
80CL580/83CL580	Low-voltage single-chip 8-bit microcontroller	811
80C592/83C592/87C592	Single-chip 8-bit microcontroller with CAN controller	851
P8xCE598	8-bit microcontroller with on-chip CAN	889
80C652/83C652	CMOS single-chip 8-bit microcontroller	916
87C652	CMOS single-chip 8-bit microcontroller	934
83C654	CMOS single-chip 8-bit microcontroller	954
87C654	CMOS single-chip 8-bit microcontroller	972
80CE654/83CE654	CMOS single-chip 8-bit microcontroller with Electromagnetic Compatibility improvements	992
83C750/87C750	CMOS single-chip 8-bit microcontrollers	1006
83C751/87C751	CMOS single-chip 8-bit microcontroller	1016
83C752/87C752	CMOS single-chip 8-bit microcontroller with A/D, PWM	1034
83CL781/83CL782	Low-voltage single-chip 8-bit microcontrollers	1052
80C851/83C851	CMOS single-chip 8-bit microcontroller with on-chip EEPROM	1085
83C852	Secured 8-bit microcontroller	1100
<b>Section 4 – Inter-Integrated (I<sup>2</sup>C) Circuit Bus</b>		
The I <sup>2</sup> C-bus and how to use it (including specification)		1141
I <sup>2</sup> C-bus peripheral selection guide		1160
82B715	I <sup>2</sup> C bus extender	1162
<b>Section 5 – ACCESS.bus Technical Overview</b>		
<b>ACCESS.bus Technical Overview</b>		1169
Introduction		1169
What is ACCESS.bus?		1169
ACCESS.bus Hardware		1169
ACCESS.bus Protocols		1170
HOW ACCESS.bus Works		1171
Electrical		1171
Bus Transactions		1171
Synchronization		1173
Byte Framing and Acknowledgement		1173
Addressing		1173
Arbitration		1173
Message Format		1173
Control/Status Messages		1173
Configuration		1174
Device Identifiers		1175
Device Capabilities Information		1175
Application Device Types		1176
Keyboard Devices		1176
Locator Devices		1176
Text Devices		1176
Timing Rules		1176
Transaction Timing Rules		1176
Response Timeouts		1176
Software Architecture and Development		1176
Device Firmware Development		1177
Host Software Architecture		1177
Development Support		1177
ACCESS.bus Industry Group		1177
Philips Semiconductors Support		1177
ACCESS.bus development kit		1180
ACCESS.bus PC/AT controller board		1182

## CONTENTS

## CONTENTS

<b>Section 6 – Control Area Network (CAN) Bus</b>		
Control Area Network (CAN) overview .....		1185
82C150	CAN serial linked I/O device with digital and analog port functions .....	1186
82C200	Stand-alone CAN-controller .....	1207
82C250	CAN controller interface .....	1243
<b>Section 7 – Other Microcontrollers</b>		
8031AH/8051AH	Single-chip 8-bit microcontroller .....	1257
8032AH/8052AH	Single-chip 8-bit microcontroller .....	1266
<b>Section 8 – Development Support Tools</b>		
Development support tools .....		1276
DB-51 CEIBO Development Board .....		1285
DS-51 CEIBO In-Circuit Emulator .....		1287
DS-750 CEIBO Emulator .....		1295
DS-752 CEIBO Emulator .....		1297
MP-51 CEIBO Programmer .....		1299
MetaLink iceMASTER-PE 8051 Family In-Circuit Emulator .....		1301
MetaLink iceMASTER 8051 Family In-Circuit Emulators .....		1305
NOHAU EMUL51-PC PC-Based In-Circuit Emulator .....		1319
LPC 80C51 Family Programmers .....		1329
PD551 Development System for 80C51 and Derivatives .....		1331
S87C00KSD 80C51 – I <sup>2</sup> C Bus Evaluation Board .....		1335
OM4129 Symbolic Debugging Package XRAY51 for the SDS 8051 Emulator .....		1336
OM4136 8051 C Cross-Compiler .....		1339
OM4142 (ASM51) Cross Assembler Package for 80C51/8051-based Systems .....		1341
OM4144 (PLMTI51) PL/M-51 Compiler Package for 80C51/8051-based Systems .....		1343
SDS 8051 Standalone Debug Station for 80C51/8051-based Systems .....		1346
<b>Section 9 – Package Outlines</b>		
SOT96A	8-Pin Plastic SOL (Small Outline Large) Dual In-Line (D/T) Package .....	1355
SOT97	8-Pin Plastic Dual In-Line (N/P) Package .....	1356
SOT101/0410D	24-Pin (300 mils wide) Plastic Dual In-Line (N) Package .....	1357
SOT117	28-Pin Plastic Dual In-Line (N/P) Package .....	1358
SOT129	40-Pin Plastic Dual In-Line (N/P) Package .....	1359
SOT136A	28-Pin Plastic SO (Small Outline) Dual In-Line (D) Package .....	1360
SOT158A	40-Pin Plastic VSO (Very Small Outline) Dual In-Line (D/T) Package .....	1361
SOT187	44-Pin Plastic Leaded Chip Carrier; Pocket Version (A) Package .....	1362
SOT188AA	68-Pin Plastic Leaded Chip Carrier; Pocket Version (A) Package .....	1363
SOT190	56-Pin Plastic VSO (Very Small Outline) Dual In-Line (D/T) Package .....	1364
SOT205	44-Pin Plastic Quad Flat Pack (B) Package .....	1365
SOT261/0401F	28-Pin (300 mils wide) Plastic Leaded Chip Carrier (A) Package .....	1366
SOT270AG	42-Lead Plastic Shrink Dual In-Line Package .....	1367
SOT274	64-Pin Plastic Shrink Dual In-Line Package .....	1368
SOT307-2	44-Pin Plastic Quad Flat Pack (B) Package .....	1369
SOT311	44-Pin Square Plastic Quad Flat Pack (B) Package .....	1370
SOT318	80-Pin Plastic Quad Flat Pack (B) Package .....	1371
SOT319	64-Pin Plastic Quad Flat Pack Package .....	1372
SOT340	24-Pin Plastic Shrink Small Outline Package (SSOP) .....	1373
SOT341	28-Pin Plastic Shrink Small Outline Package (SSOP) .....	1374
SOT351	80-Lead Ceramic Chip Carrier (CQFP) .....	1375
SOT389	(refer to package number 1711–)	
0398E	68-Pin Plastic Leaded Chip Carrier (A) Package .....	1376
0401F/SOT261	28-Pin (300 mils wide) Plastic Leaded Chip Carrier (A) Package .....	1366
0403G	44-Pin Plastic Leaded Chip Carrier (A) Package .....	1377
0410D/SOT101	24-Pin (300 mils wide) Plastic Dual In-Line (N) Package .....	1357
0413B	28-pin (600 mils wide) Plastic Dual In-Line (N) Package .....	1378
0414B	64-Pin (900 mils wide) Plastic Dual In-Line (N) Package .....	1379
0415C	40-Pin (600 mils wide) Plastic Dual In-Line (N) Package .....	1380



## CONTENTS

0586B	24-Pin (300 mils wide) Ceramic Dual In-line (F) Package (with Window (FA) Package)	1381
0589B	28-Pin (600 mils wide) Ceramic Dual In-line (F) Package (with Window (FA) Package)	1382
0590B	40-Pin (600 mils wide) Ceramic Dual In-line (F) Package (with Window (FA) Package)	1383
1118D	44-Pin Plastic Quad Flat Pack (B) (MEC) Package	1384
1240C	68-Pin Chip Carrier, J-Bend (L) Package	1385
1472A	44-pin CerQuad J-Bend (K) Package	1386
1473A	68-pin CerQuad J-Bend (K) Package	1387
1680	42-Pin Plastic SSOP (Shrink Small Outline Package) Dual In-Line (D/K) Package	1388
1711 -	44-Pin Plastic Thin Quad Flat Pack Package (comparable to SOT389)	1389

Section 10 – Sales Offices, Representatives, and Distributors ..... 1393Appendix A – Data Handbook System ..... 1394Appendix B – Pin Configurations ..... 1396

[illegible]

## 80C51 microcontroller family features guide

	Part Number (ROMless)	Memory			Counter Timers	I/O Port	Serial Interfaces	External Interrupt	Comments/ Special Features
		ROM	EPRM	RAM					
P	83C750	1K		64	1 (16-bit)	2-3/8	-	2	40 MHz, Lowest cost, 24-pin Skinny DIP
P	87C750		1K	64	1 (16-bit)	2-3/8	-	2	40 MHz, Lowest cost, 24-pin Skinny DIP
P	83C751	2K		64	1 (16-bit)	2-3/8	I2C (bit)	2	Low-Cost 24-pin Skinny DIP
P	87C751		2K	64	1 (16-bit)	2-3/8	I2C (bit)	2	Low-Cost 24-pin Skinny DIP
P	83C752	2K		64	1 (16-bit)	2-5/8	I2C (bit)	2	5 Channel 8-bit A/D, PWM Output
P	87C752		2K	64	1 (16-bit)	2-5/8	I2C (bit)	2	5 Channel 8-bit A/D, PWM Output
MAX	8051AH (8031AH)	4K		128	2	4	UART	2	NMOS
SC	SC80C51 (80C31)	4K		128	2	4	UART	2	CMOS (Sunnyvale)
PC	PCx80C51 (80C31)	4K		128	2	4	UART	2	CMOS (Hamburg)
SC	87C51		4K	128	2	4	UART	2	CMOS
P	80CL51 (80CL31)	4K		128	2	4	UART	10	Low Voltage (1.8V to 6V), Low Power
P	83CL410 (80CL410)	4K		128	2	4	I2C	10	Low Voltage (1.8V to 6V), Low Power
SC	83C451 (80C451)	4K		128	2	7	UART	2	Extended I/O, Processor Bus Interface
SC	87C451		4K	128	2	7	UART	2	Extended I/O, Processor Bus Interface
P	83C550 (80C550)	4K		128	2 + Watchdog	4	UART	2	8 Channel 8-bit A/D
P	87C550		4K	128	2 + Watchdog	4	UART	2	8 Channel 8-bit A/D
P	83C851 (80C851)	4K		128	2	4	UART	2	256B EEPROM, 80C51 Pin compatible
P	83C852	6K		256	2 (16-bit)	2/8	-	1	Smartcard Controller with 2K EEPROM (Data, Code) Cryptographic Calc Unit
P	83CL580	6K		256	3 + Watchdog	5	UART, I2C	10	4 Channel 8-bit A/D, PWM Output, Low Voltage (2.5V to 6V), Low Power
MAX	8052AH (8032AH)	8K		256	3	4	UART	2	NMOS
P	80C52 (80C32)	8K		256	3	4	UART	2	80C51 Pin Compatible
P	87C52		8K	256	3	4	UART	2	(see above)
P	83C652 (80C652)	8K		256	2	4	UART, I2C	2	80C51 Pin Compatible
S	87C652		8K	256	2	4	UART, I2C	2	(see above)
S	83C51FA	8K		256	3 + PCA	4	UART	2	Enhanced UART, 3 timers + PCA
S	87C51FA		8K	256	3 + PCA	4	UART	2	Enhanced UART, 3 timers + PCA
S	83L51FA	8K		256	3 + PCA	4	UART	2	Low Voltage 83C51FA (3V @ 20MHz)
S	87L51FA		8K	256	3 + PCA	4	UART	2	Low Voltage OTP 87C51FA (3V @ 20MHz)
P	83C575 (80C575)	8K		256	3 + PCA + Watchdog	4	UART	2	High Reliability, with Low Voltage Detect, Osc Fail Detect, Analog Comparators, PCA
P	87C575		8K	256	(see above)	4	UART	2	(see above)
PC	83C552 (80C552)	8K		256	3 + Watchdog	6	UART, I2C	2	8 Channel 10-bit A/D, 2 PWM Outputs, Capture/Compare Timer
S	87C552		8K	256	3 + Watchdog	6	UART, I2C	2	(see above)
PC	83C562 (80C562)	8K		256	3 + Watchdog	6	UART	2	8 Channel 8-bit A/D, 2 PWM Outputs, Capture/Compare Timer
P	83C055	16K		256	2 (16-bit)	3.5	-	2	On-Screen Display, 9 PWM Outputs, 3 Software A/D Inputs
P	87C055		16K	256	2 (16-bit)	3.5	-	2	(see above)
P	83C654	16K		256	2	4	UART, I2C	2	80C51 Pin Compatible
P	87C654		16K	256	2	4	UART, I2C	2	(see above)
P	83CE654	16K		256	2	4	UART, I2C	2	83C654 with Reduced EMI
P	83CL781	16K		256	3	4	UART, I2C	10	Low Voltage (1.8V to 6V), Low Power
P	83CL782	16K		256	3	4	UART, I2C	10	83CL781 Optimized 12MHz @ 3.1V
S	83C51FB	16K		256	3 + PCA	4	UART	2	Enhanced UART, 3 timers + PCA
S	87C51FB		16K	256	3 + PCA	4	UART	2	Enhanced UART, 3 timers + PCA
S	83L51FB	16K		256	3 + PCA	4	UART	2	Low Voltage 83C51FB (3V @ 20MHz)
S	87L51FB		16K	256	3 + PCA	4	UART	2	Low Voltage OTP 87C51FB (3V @ 20MHz)
P	83C524	16K		512	3 + Watchdog	4	UART, I2C-bit	2	512 RAM
P	87C524		16K	512	3 + Watchdog	4	UART, I2C-bit	2	512 RAM
P	83C592 (80C592)	16K		512	3 + Watchdog	6	UART, CAN	6	CAN Bus Controller with 8 x 10-bit A/D, 2 PWM outputs, Capture/Compare Timer
P	87C592		16K	512	3 + Watchdog	6	UART, CAN	6	(see above)
P	83C528 (80C528)	32K		512	3 + Watchdog	4	UART, I2C-bit	2	Large Memory for High Level Languages
P	87C528		32K	512	3 + Watchdog	4	UART, I2C-bit	2	Large Memory for High Level Languages
P	83CE558(80CE558)	32K		1024	3 + Watchdog	6	UART, I2C	2	Low EMI, 8 Channel 10-bit A/D, 2 PWM Outputs, Capture/Compare Timer
P	89CE558		32K	1024	3 + Watchdog	6	UART, I2C	2	32K Flash EEPROM plus above

Note: Part number prefixes are noted in the first column. All combinations of part type, speed, temperature and package may not be available.

## 80C51 microcontroller family features guide

Part Number (ROMless)	Program Security?	Clock Freq (MHz)	Temperature Ranges (°C)			Package				
			0 to 70	-40 to +85	-55 to +125	PDIP	CDIP	PLCC	CLCC	PQFP
83C750	S	Y	3.5 to 40	X	X	N24	F24	A28		
87C750	S	Y	3.5 to 40	X	X	N24	F24	A28		
83C751	S	N	3.5 to 16	X	X	N24		A28		
87C751	S	Y	3.5 to 16	X	X	N24	F24	A28		
83C752	S	N	3.5 to 16	X	X	N28		A28		
87C752	S	Y	3.5 to 16	X	X	N28	F28	A28		
8051AH (8031AH)	S	N	3.5 to 15	X	X	N40		A44		
SC80C51 (80C31)	S	Y	3.5 to 33	X	X	N40		A44		B44 (5)
PCx80C51 (80C31)	H	N	1.2 to 30	X	X	P (40)		WP (44)		H (44)
87C51	S	Y	3.5 to 33	X	X	N40	F40	A44	K44	B44 (5)
80CL51 (80CL31)	Z	N	0 to 16 (1)	X		N40 (2)				B44
83CL410 (80CL410)	Z	N	0 to 16 (1)	X		N40 (2)				
83C451 (80C451)	S	N	3.5 to 16	X	X	N64 (4)		A68		
87C451	S	Y	3.5 to 16	X	X	N64 (4)		A68		
83C550 (80C550)	S	Y	3.5 to 16	X	X	N40		A44		
87C550	S	Y	3.5 to 16	X	X	N40	F40	A44	K44	
83C851 (80C851)	H	Y	1.2 to 16	X	X	N40		A44		B44
83C852	H	Y	1 to 12	X		SO28 or die				
83CL580	Z	N	0 to 12 (1)		X	(3)				B64
8052AH (8032AH)	S	N	3.5 to 15	X	X	N40		A44		
80C52 (80C32)	S	Y	3.5 to 24	X	X	N40		A44		B44 (5)
87C52	S	Y	3.5 to 24	X	X	N40	F40	A44	K44	B44 (5)
83C652 (80C652)	H	Y	1.2 to 24	X	X	N40		A44		B44
87C652	S	Y	1.2 to 20	X	X	N40	F40	A44	K44	
83C51FA	S	Y	3.5 to 24	X	X	N40	F40	A44	K44	B44
87C51FA	S	Y	3.5 to 24	X	X	N40	F40	A44	K44	B44
83L51FA	S	Y	3.5 to 20	X	X	N40	F40	A44	K44	B44
87L51FA	S	Y	3.5 to 20	X	X	N40	F40	A44	K44	B44
83C575 (80C575)	S	Y	4 to 16	X		N40		A44		B44
87C575	S	Y	4 to 16	X	X	N40	F40	A44	K44	B44
83C552 (80C552)	H	N	1.2 to 30	X	X			A68		B80
87C552	S	Y	1.2 to 16	X				A68	K68	
83C562 (80C562)	H	N	1.2 to 16	X	X			A68		B80
83C055	S	N	3.5 to 20	X		42SDIP				
87C055	S	N	3.5 to 20	X		42SDIP				
83C654 (80C654)	H	Y	1.2 to 24	X	X	42SDIP	N40	A44		B44
87C654	S	Y	1.2 to 20	X	X	N40	F40	A44	K44	B44
83CE654	H	Y	1.2 to 16	X	X					B44
83CL781	Z	N	0 to 12 (1)		X	N40				B44
83CL782	Z	N	0 to 12 (1)		-25 to +55	N40				B44
83C51FB	S	Y	3.5 to 24	X	X	N40	F40	A44	K44	B44
87C51FB	S	Y	3.5 to 24	X	X	N40	F40	A44	K44	B44
83L51FB	S	Y	3.5 to 20	X		N40	F40	A44	K44	B44
87L51FB	S	Y	3.5 to 20	X		N40	F40	A44	K44	B44
83C524	H	Y	1.2 to 16	X	X	N40		A44		B44
87C524	S	Y	3.5 to 12	X	X	N40	F40	A44	K44	B44
83C592 (80C592)	H	Y	1.2 to 16		X			A68	K68	
87C592	H	Y	1.2 to 16	X		42SDIP		A68	K68	
83C528 (80C528)	H	Y	1.2 to 16	X	X	N40		A44		B44
87C528	S	Y	3.5 to 20	X	X	N40	F40	A44	K44	B44
83CE558 80CE558	H	Y	3.5 to 16	X	X					B80
89CE558	H	Y	3.5 to 16	X	X					B80

Note: Production Centers are indicated in the second column: H - Hamburg, S - Sunnyvale, Z - Zurich. 1) Oscillator options start from 32kHz. 2) Also available in VSO40 package. 3) Also available in VSO56 Package. 4) Not recommended for new design. 5) Package available up to 16 MHz only.



## 8051 microcontroller cross-reference guide

	INTEL	AMD	SIEMENS	OKI	MATRA/HARRIS	PHILIPS SEMICONDUCTORS
NMOS	8039AL			X	X	MAB8039H/SCN8039H
	8049AH			X	X	MAB8049H/SCN8049H
	8040AHL		X	X	X	MAB8040H/SCN8040H
	8050AH		X	X	X	MAB8050H/SCN8050H
				X	X	
				X	X	
	8031AH	8031AH	SAB 8031A	X	X	MAB8031AH/SCN8031H
	8051AH	8051AH	SAB 8051A	X	X	MAB8051AH/SCN8051H
	8032AH		SAB 8032A	X	X	MAB8032AH/SCN8032H
	8052AH		SAB 8052A	X	X	MAB8052AH/SCN8052H
CMOS	80C31BH	80C31BH	SAB 80C31	MSM80C31	80C31	PCB80C31BH-2/SC80C31BCC
	80C31BH-1				80C31-1	PCB80C31BH-3/SC80C31BCG
	80C31BH-2		C51	MSM80C31	80C3151	/SC80C31BCB
	80C51BH	8051BH	SAB 80C51	MSM80C51	80C51	PCB80C51BH-2/SC80C51BCC
	80C51BH-1				80C51-1	PCB80C51BH-3/SC80C51BCG
	80C51BH-2			MSM80C51	80C51	/SC80C51BCB
	87C51	87C51				SC87C51CC
	87C51-1			X	X	SC87C51CG
	87C51-2			X	X	SC87C51CB
			X	X	X	
	80C32			X	X	P80C32EB
	80C32-1		X	X	X	P80C32GB
	80C52	80C52T2		X	X	P80C52EB
	80C52-1			X	X	P80C52GB
				X	X	
	83C51FA		X		X	S83C51FA-4
	87C51FA					S87C51FA-4
	83C51FB		X	X	X	S83C51FB-4
	87C51FB			X	X	S87C51FB-4
	83C51FC				X	S83C51FC-4
	87C51FC			X	X	S87C51FC-4

## NOTES:

1. Siemens 8032A-16 = 16MHz 8032.
2. AMD 80C52T2 = 80C52 without T2.
3. 80XXAHL = 80XX with low power standby pin; H = HMOS.

## Low Power 80CL51 Family Derivatives

## 83CLXXX Microcontroller Family

### QUICK REFERENCE CHART

Type	Available	ROM	RAM	I/O	I <sup>2</sup> C	UART	Features	Package
80CL51	Yes	4k	128	32	No	Yes	Low Voltage 80C51	40-Pin Dual In-Line 40-Pin Very Small Outline 44-Pin Quad Flat Pack
80CL31	Yes	—	128	32	No	Yes	Low Voltage 80C31	40-Pin Dual In-Line 40-Pin Very Small Outline 44-Pin Quad Flat Pack
83CL410	Yes	4k	128	32	Yes	No	80CL51 with I <sup>2</sup> C-bus	40-Pin Dual In-Line 40-Pin Very Small Outline 44-Pin Quad Flat Pack
80CL410	Yes	—	128	32	Yes	No	80CL51 with I <sup>2</sup> C-bus	40-Pin Dual In-Line 40-Pin Very Small Outline 44-Pin Quad Flat Pack
83CL580	Yes	6k	256	40	Yes	Yes	ADC, PWM, Watchdog, T2	50-Pin Very Small Outline 64-Pin Quad Flat Pack
80CL580	Yes	—	256	40	Yes	Yes	ADC, PWM, Watchdog, T2	50-Pin Very Small Outline 64-Pin Quad Flat Pack
83CL781	Yes	16k	256	32	Yes	Yes	Low voltage 83C654, T2	40-Pin Dual In-Line 44-Pin Quad Flat Pack
83CL782	Yes	16k	256	32	Yes	Yes	Fast 83CL781: 12MHz/3V	40-Pin Dual In-Line 44-Pin Quad Flat Pack
85CL000	Yes	—	256	32	Yes	Yes	For SW development	Piggyback
85CL580	Yes	—	256	40	Yes	Yes	For SW development	Piggyback
85CL782	Yes	—	256	32	Yes	Yes	For SW development	Piggyback

### LOW VOLTAGE OTP DEVICES

Type	Available	ROM	RAM	I/O	I <sup>2</sup> C	UART	Features	Package
83L51FA	Yes	8k	256	32	No	Yes	PCA, Enhanced UART 3.0V to 4.5V	40-Pin Dual In-Line 44-Pin PLCC 44-Pin Quad Flat Pack
87L51FA	Yes	8k EPROM/ OTP	256	32	No	Yes	PCA, Enhanced UART 3.0V to 4.5V	40-Pin Dual In-Line 44-Pin PLCC 44-Pin Quad Flat Pack
83L51FB	Yes	16k	256	32	No	Yes	PCA, Enhanced UART 3.0V to 4.5V	40-Pin Dual In-Line 44-Pin PLCC 44-Pin Quad Flat Pack
83L51FB	Yes	16k EPROM/ OTP	256	32	No	Yes	PCA, Enhanced UART 3.0V to 4.5V	40-Pin Dual In-Line 44-Pin PLCC 44-Pin Quad Flat Pack

## 80C51 microcontroller development system support

## DEVELOPMENT SYSTEM CONTACTS

COMPANY	ADDRESS	TELEPHONE
Ashling Microsystems Limited	Plassey Technological Park Limerick, Ireland	(353) 63-334466
BSO Tasking	128 Technology Center P.O. Box 9164 Waltham, MA 02254-9164	(617) 894-7800
Ceibo Ltd.	105 Gleason Rd. Lexington, MA 02173	(617) 863-9927
	Merkazim Building, Industrial Zone P.O. Box 2106 Herzella 46120, ISRAEL	972-52-555387
MetaLink Corp.	325 E. Elliot Road, Suite 23 Chandler, AZ 85225	(602) 926-0797
Nohau Corp.	51 E. Campbell Ave. Campbell, CA 95008	(408) 866-1820
Philips Semiconductors	Corporate Centre Building BAE-2 P.O. Box 218 5600 MD Eindhoven The Netherlands	31-40-724223
SIGNUM Systems	171 E. Thousand Oaks Blvd., #202 Thousand Oaks, CA 91360	(805) 371-4608

## EPROM PROGRAMMING SUPPORT CONTACTS

Advin Systems 1050-L East Duane Ave. Sunnyvale, CA 94086 (408) 736-2503	Logical Devices, Inc. 1201 Northwest 65th Place Ft. Lauderdale, FL 33309 (305) 974-0967	North Valley Products P.O. Box 32899 San Jose, CA 95152 (408) 929-5345
BP Microsystems 10681 Haddington #190 Houston, TX 77043 (800) 225-2102, (713) 461-9430	Logical Systems P.O. Box 6184 Syracuse, NY 13217-6184 (315) 478-0722	Strebtor Data Communications 1008 N. Nob Hill American Fork, UT 84003 (801) 756-3605
Data I/O Corp. 10525 Willows Road N.E. P.O. Box 97046 Redmond, WA 98073-9746 (206) 881-6444	Needham's Electronics 4535 Orange Grove Ave. Sacramento, CA 95841 (916) 924-8037	

## SOFTWARE SUPPORT CONTACTS

COMPANY	ADDRESS	TELEPHONE
Archimedes Software, Inc.	2159 Union St. San Francisco, CA 94123	(415) 567-4010
BSO/Tasking	Tasking Software BV P.O. Box 899 3800 AW Amersfoort The Netherlands  BSO Tasking 128 Technology Center P.O. Box 9164 Waltham, MA 02254-9164	31-33-55-85-84 (Telephone) 31-33-55-00-33 (Fax)  (617) 894-7800 (Telephone) (617) 894-0551 (Fax) (710) 324-0760 (Telex) (800) 458-8276 (Toll Free)
Franklin Software, Inc.	888 Saratoga Ave. #2 San Jose, CA 95129	(408) 296-8051

## NOTE:

For more information on Development Support, see Section 8 of this book.

## 8-bit microcontroller demonstration and evaluation boards

PRODUCT	DESCRIPTION
OM4151	I <sup>2</sup> C demonstration board based on 84CXX derivatives I <sup>2</sup> C LCD driver demonstration board with 84CXX microcontroller I <sup>2</sup> C bus analyzer (with 84CXX)
S87C00K	I <sup>2</sup> C demonstration board based on 80C51 derivatives
OM4238, P8051DB	8051 family demonstration board
OM4128	8XC552 evaluation board
OM4130, PCAN-EVAL	CAN controller evaluation board
OM4239	8XC592 evaluation board
OM4160, SM68070	68070 demonstration and evaluation board MicroCore I
OM4160/3, SM90C100	90C100 family demonstration and evaluation board MicroCore III
OM4280, P83C852DEM	83C852 demonstration kit
OM4281	83C852 software evaluation kit
P8051DB	80C51 family development board
OM4717, P83CLSOLAR	83CL410 solar powered demonstration board
OM5005, P80CLEVAL	80CL51 evaluation board
DS750	8XC750 microcontroller in-circuit emulation development tool



## Microcontroller bulletin boards

To better serve our customers, Philips maintains a microcontroller bulletin board. This computer bulletin board system features a microcontroller newsletter, application and demonstration programs for download, and the ability to send messages to microcontroller application engineers. The system can be accessed with a modem at 2400, 1200, or 300 baud.

The telephone numbers are:

**(800) 451-6644 (in the U.S.)**

or

**(408) 991-2406**

**Please call us anytime!**

We also have a ROM code bulletin board through which you can submit ROM codes. This is a closed bulletin board for security reasons. To get an ID, contact your local sales office. The system can be accessed with a 2400, 1200, or 300 baud modem, and is available 24 hours a day.

The telephone number is:

**(408) 991-3459**

## CMOS and NMOS 8-bit microcontroller family

## 80C51 FAMILY CMOS

TYPE	ROM/EPROM	RAM	SPEED (MHz)	PACKAGE	FUNCTIONS	REMARKS	PROBE SDS	THIRD PARTY EMULATOR	REMARKS
80C31 80C51 87C51	0 4k ROM 4k EPROM	128 128 128	33 33 33	DIL40, LCC44 QFP44	UART, 2 timers	87C51: QFP package up to 16MHz	OM1092 + OM1097 (16MHz)	8052PC(M) POD-C51B(N)	OM1092: Universal probe OM1095: Upgrade unit
83C51FA 87C51FA	8k ROM 8k EPROM	256 256	24 24	DIL40, LCC44 QFP44	Enhanced UART, 3 timers, PCA				
83L51FA 87L51FA	8k ROM 8k EPROM	256 256	20 20	DIL40, LCC44 QFP44	Enhanced UART, 3 timers, PCA	3V to 4.5V operation			
87C51FB 83C51FB	8k ROM 8k EPROM	256 256	24 24	DIL40, LCC44 QFP44	Enhanced UART, 3 timers, PCA				
87L51FB 83L51FB	8k ROM 8k EPROM	256 256	20 20	DIL40, LCC44 QFP44	Enhanced UART, 3 timers, PCA	3V to 4.5V operation			
87C51FC 83C51FC	32k ROM 32k EPROM	256 256	20 20	DIL40, LCC44	Enhanced UART, 3 timers, PCA				
80C32 80C52 87C52	0 8k ROM 8k EPROM	256 256 256	24 24 24	DIL40, LCC44 QFP44	UART, 3 timers		OM1079 OM5012	8052PC(M) POD-C32(N)	
80C451 83C451 87C451	0 4k ROM 4k EPROM	128 128 128	16 16 16	DIP64/LCC68	UART, 2 timers Extended I/O		OM4123	83C451PC(M) POD-C451B(N)	OM4124: PLCC to DIL OM4125: DIL to PLCC
83C504 87C504	16K ROM 16K EPROM	256 256	24 24	DIL40, LCC44 QFP44	24 by 8 divide, 2 timers				
87C524	16K EPROM	512	20	DIL40/LCC44	UART, 3 timers Watchdog timer Bit I <sup>2</sup> C		OM4111 + OM4110	83528PC(M) POD-C528(N)	OM4110: gen. probe OM4111: probe head
83C528 87C528	32k ROM 32k EPROM	512 512	16 16, 20	DIL40/LCC44 (QFP44)	UART, 3 timers Watchdog timer Bit I <sup>2</sup> C		OM4111 + OM4110	83C528PC(M) POD-C528(N)	OM4110: gen. probe OM4111: probe head OM4120-S for max. speed
83CE528	32kROM	512	16	CE ONLY QFP					
80C550 83C550 87C550	0 4k ROM 4k EPROM	128 128 128	16 16 16	LCC44 DIL40	UART, 2 timers 8 8-bit ADC inputs, watchdog timer		OM5055 + OM4110	83550(M) POD-C550(N)	OM4110: probe base
80C552 83C552 87C552	0 8k ROM 8k EPROM	256 256 256	16, 24 16, 24 16	LCC68/QFP80	UART, 2 timers Timer with compare and capture, 2 PWM outputs, 8 10-bit ADC inputs, Byte I <sup>2</sup> C		OM1092 + OM1095	83C552PC(M) POD-C552B(N)	OM1092: Universal probe OM1095: Upgrade unit
83CE558 87CE558 80CE558	32K ROM 32K FLASH 0	1K 1K	16 16	QFP80	As 8xC552 with PLL-oscillator Auto scan ADC	89C: Q4-92 83C: Q2/3-93	OM4110 + OM4271 (in dev)		OM4110: probe base OM4115: QFP80 adapter

M = Metlink  
N = Nohau

## CMOS and NMOS 8-bit microcontroller family

## 80C51 FAMILY CMOS (Continued)

TYPE	ROM/ EPROM	RAM	SPEED (MHz)	PACKAGE	FUNCTIONS	REMARKS	PROBE SDS	THIRD PARTY EMULATOR	REMARKS
80C562 83C562	0 8k ROM	256 256	16 16	LCC68/QFP80	UART, 2 timers Timer with compare and capture, 2 PWM outputs, 8 8-bit ADC inputs		OM1092 + OM1095	83C552PC(M)  POD-C552B(N)	OM1092: Universal probe OM1095: Upgrade unit
80C575 83C575 87C575	0 8k 8k EPROM	256 256 256	16 16 16	DIL40, LCC44 QFP44	3 timers 1 Enh. UART, PCA, 4 analog comparators				
83C576 87C576	8k ROM 8k EPROM	256 256	16 16	DIL40, LCC44, SDIL42	10-bit A/D, 3 timers, PCA, Watchdog timer				
80C592 83C592 87C592	0 16k ROM 16k EPROM	512 512 512	16 16 16	LCC68/QFP80	8XC552 + CAN interface		OM4110 + OM4112	POD-592(N)	OM4110: gen. probe OM4112: probe head OM4120S: full speed
87CE598 87CE598 80CE598	32K ROM 32k EPROM 0	512 512 512	16 16 16	QFP80	8xC552 + CAN interface. No I <sup>2</sup> C	87CE: prod: Q2'94	OM4110 + OM4114 OM4115		OM4110: probe base OM4115: QFP adapter
80C652 83C652 87C652	0 8k ROM 8k EPROM	256 256 256	16, 24 16, 24 16, 20	DIL40/LCC44 QFP44	UART, 2 timers Byte I <sup>2</sup> C		OM1092 + OM1096	83652PC(M) POD-C51B(N)	
83C654 87C654	16k ROM 16k EPROM	256 256	16, 24 16, 20	DIL40/LCC44 QFP44	UART, 2 timers Byte I <sup>2</sup> C		OM1092 + OM1096	83654(M)  POD-C51B(N)	OM1092: Universal probe OM1095: Upgrade unit
83CE654 80CE654	16k ROM 0	256 256	16 16	QFP44	UART, 2 timers Byte I <sup>2</sup> C	83C654 with Electromagnetic Compatibility improvements	OM1092 + OM1096	83654(M)  POD-C51B(N)	OM1092: Universal probe OM1095: Upgrade unit
83C750 87C750	1K ROM 1KEPROM	64 64	40 40	SDIP24 skinny	1 timer		OM1094		Up to 16MHz
83C751 87C751	2k ROM 2k EPROM	64 64	16 16	DIP24 skinny LCC28 DIP24 skinny	1 timer Bit I <sup>2</sup> C		OM1094P	83751PC(M) POD-C751(N)	
83C752	2k ROM	64	16	DIP28, LCC28	1 timer, PWM output, 5 8-bit ADC inputs, Bit I <sup>2</sup> C		OM5072	83752A(M)	
83C752	2k EPROM	64	16	DIP 28, LCC28				POD-C752(N)	
80C851 83C851	0 4k ROM	128 128	16 16	DIL40/LCC44 QFP44	UART, 2 timers 256 byte		OM1092	80851PC(M) POD-C51(N)	
83C852	6k ROM	256	6		2k byte EEPROM smart card hardware CU		OM4119		
83C055 87C055	16k ROM 16k EPROM	256 256	12 12	DIP42 Shrunk DIP42 Shrunk	As 8XC053	In dev.	OM5054		

\* The following microcontrollers have no external memory access: 8XC751, 8XC752, 8XC053, 87C054, 83C852.

M = Metlink  
N = Nohau

## CMOS and NMOS 8-bit microcontroller family

## 80CLXXX FAMILY CMOS

TYPE	ROM	RAM	SPEED (MHz)	PACKAGE	FUNCTIONS	REMARKS	PROBE SDS	REMARKS
85CL000	0	256	12	Piggyback	Piggyback CL410, CL411, CL51, P80C51			
85CL580	0	256	12	Piggyback	Piggyback CL580			
85CL781	0	256	12	Piggyback	Piggyback CL781, CL782, CL52			
80CL51 80CL31	4K 0	128 128	12 12	DIL40 VSO40	2 timers, UART		OM1079	QFP: OM5020
83CL410 80CL410	4k 0	128 128	12 12	DIL40 VSO40	2 timers Byte I <sup>2</sup> C		OM1079	QFP: OM5020
83CL580	6k	256	16	QFP64/ VSO56	3timers, UART Watchdog timer Byte I <sup>2</sup> C, 1 PWM 4*8 bit ADC		OM1079 + OM5004	OM1079: Probe base OM5004: Probe adap
83CL781 83CL782	16k 16k	256 256	12 @ 4.5V 12 @ 3V	DIL40 QFP44	3timers, UART Byte I <sup>2</sup> C		OM1079 + OM5004 + tbd	OM1079: Probe base OM5004: Probe adap
83CL167 83CL267	16K 12K	256 256	12 12	SDIL64 QFP64	3timers 1-14 bit PWM 4-6 bit PWM 4-7 bit PWM 4*4 bit ADC Byte I <sup>2</sup> C 160 char OSD 126 char fonts 4 char sizes Shadow modes ODS PLL osc. 10MHz Blinking	In Dev	OM4840 OM1079	
83CL168 83CL268	16K 12K	256 256	12 12	SDIL64 QFP64	3timers 1-14 bit PWM 4-6 bit PWM 4-7 bit PWM 4*4 bit ADC RC preprocessor Byte I <sup>2</sup> C 3 wire serial I/O 160 char OSD 126 char fonts 4 char sizes Shadow modes ODS PLL osc. 10MHz Blinking	In Dev	OM4840 + OM1079	

## CMOS and NMOS 8-bit microcontroller family

## 8051 FAMILY NMOS

[illegible]



## CMOS and NMOS 8-bit microcontroller family

## 8400 FAMILY CMOS

(continued)

TYPE	ROM	RAM	SPEED (MHz)	PACKAGE	FUNCTIONS	REMARKS	PROBE SDS	REMARKS
84C21A	2k	64	10	DIL28/SO28	20 I/O lines		OM1083	OM1025
84C41A	4k	128	10	DIL28/SO28	8-bit timer			(LSDS)
84C81A	8k	256	10	DIL28/SO28	Byte I <sup>2</sup> C			
84C22A	2k	64	10	DIL20/SO20	13 I/O lines		OM1083 +	OM1025
84C42A	4k	64	10	DIL20/SO20	8-bit timer		Adapter_1	(LSDS)
84C12A	1k	64	16	DIL20/SO20				
84C00B	0	256	10	28 pins	20 I/O lines	Piggyback	OM1080	
84C00T	0	256	10	VSO-56	8-bit timer	ROMless	OM1080	
84C121	1k	64	10	DIL20/SO20	Byte I <sup>2</sup> C			
84C121B	0	64	10		13 I/O lines	Piggyback	OM1073	OM1025(LEDs)
84C122A	1k	32	10		2 8-bit timers			
84C122B	4K	32			8 bytes			OM1027
84C422A					EEPROM			
84C422B	8K	32		A: SO20	Controller for		OM4830	
84C822A				B: SO24	remote control			
84C822B				C: SO28	A: 12 I/O			
84C822C					B: 16 I/O			
					C: 20 I/O			
84C230	2l	64	10	DIL40/VSO40	12 I/O lines		OM1072	
					8-bit timer			
					16*4 LCD drive			
84C430	4k	128	10	QFP64	24 I/O lines		OM1072	
					8-bit timer			
					Byte I <sup>2</sup> C			
84C430BH	0	128	10		24*4 LCD drive	Piggyback for C230 and C430		
84C633	6k	256	16	VSO56	28 I/O lines		OM1086	
					8-bit timer			
					16-bit up/down counter			
					16-bit timer with compare and capture			
84C633B	0	256	16		16*4 LCD drive			
84C440	4k	128	10	DIP42 shrunk	RC: 29 I/O lines	I <sup>2</sup> C, RC	OM1074	For emulation of LC versions, use OM1074 + adapter_3 + 2 adapter_5
84C441	4k	128	10		LC: 28 I/O lines	I <sup>2</sup> C, LC		
84C443	4k	128	10		8-bit timer	RC		
84C444	4k	128	10		1 14-bit PWM	LC		
84C640	6k	128	10		5 6-bit PWM	I <sup>2</sup> C, RC		
84C641	6k	128	10		3-bit ADC	I <sup>2</sup> C, LC		
84C643	6k	128	10		OSD 2L-16	RC		
84C644	6k	128	10			LC		
84C840	8k	192	10			I <sup>2</sup> C, RC		
84C841	8k	192	10			I <sup>2</sup> C, LC		
84C843	8k	192	10			RC		Baud for LCDS OM4831
84C844	8k	192	10			LC		

## CMOS and NMOS 8-bit microcontroller family

## 8400 FAMILY CMOS (Continued)

TYPE	ROM	RAM	SPEED (MHz)	PACKAGE	FUNCTIONS	REMARKS	PROBE SDS	REMARKS
84C646 84C846	6k 8k	192 192	10 10	DIP42 shrunk	30 I/O lines DOS clock = PLL 8 bit timer 1-14 bit PWM 4-6 bit PWM 4-7 bit PWM 3-4 bit ADC DOS: 64 disp. RAM 62 char. fonts Char. blinking Shadow modes 8 foreground colors/char. 8 background colors/word DOS: clock: 8...20MHz	I <sup>2</sup> C, RC I <sup>2</sup> C, RC	OM4829 + OM4832	OM4833 for LCD584A
84C85	8k	256	10	DIL40/VSO40	32 I/O lines 8-bit timer Byte I <sup>2</sup> C		CM1070	
84C85B	0	256	10			Piggyback for C85		
84C853	8k	256	16	DIL40/VSO40	33 I/O lines 8-bit timer 16-bit up/down counter 16-bit timer with compare and capture		OM1081	
84C853B	0	256	16			Piggyback for C853		
84C270 84C470	2k 4k	128 128	10 10	DIL40/VSO40 DIL40/VSO40	8 I/O lines 16*8 capture keyboard matrix 8-bit timer		OM1077	
84C270B	0	128	10			Piggyback for C270		
84C470B	0	128	10		470 also handles mech. keys	Piggyback for C470		
84C271	2k	128	10	DIL40	8 I/O lines 16*8 mech. keyboard matrix 8-bit timer		OM1078	

## 8400 FAMILY NMOS

TYPE	ROM	RAM	SPEED (MHz)	PACKAGE	FUNCTIONS	REMARKS	EMULATOR TOOLS	REMARKS
8411 8421 8441 8461	1k 2k 4k 6k	64 64 128 128	6 6 6 6	DIL28/SO28 DIL28/SO28 DIL28/SO28 DIL28/SO28	20 I/O lines 8-bit timer Byte I <sup>2</sup> C		OM1084	OM1025 (LCDS) +
8422 8442	2k 4k	64 128	6 6	DIL20 DIL20	13 I/O lines 8-bit timer Bit I <sup>2</sup> C		PM8327/20 + PM8447	On PMDS
8401B 8401WP	0 0	128 128	6 6	28-pin PLCC68		Piggyback for 84X1 Bond out		

## CMOS and NMOS 8-bit microcontroller family

## 3300 FAMILY CMOS

(Continued)

TYPE	ROM	RAM	SPEED (MHz)	PACKAGE	FUNCTIONS	REMARKS	PROBE SDS	REMARKS
3315A	1.5k	160	10	DIL28/SO28	20 I/O lines 8-bit timer $V_{DD} > 1.8V$		OM1083	OM1025(LCDS)
3343	3k	224	10	DIL28/SO28	20 I/O lines 8-bit timer $V_{DD} > 1.8V$ Byte I <sup>2</sup> C		OM1083	OM1025(LCDS)
3344A	2k	224	3.58	DIL28/SO28	20 I/O lines 8-bit timer DTMF generator		OM1071	OM1025(LCDS) + OM1028
3346A	4k	128	10	DIL28/SO28	20 I/O lines 8-bit timer Byte I <sup>2</sup> C 256 bytes EEPROM $V_{DD} < 1.8V$		OM1076	
3347	1.5k	64	3.58	DIL20/SO20	12 I/O lines 8-bit timer DTMF generator		OM1071 + Adapter_2	OM1025(LCDS) + OM1028
3348A	8k	256	10	DIL28/SO28	20 I/O lines 8-bit timer Byte I <sup>2</sup> C $V_{DD} < 1.8V$		OM1083	OM1025(LCDS)
3349A	4k	224	3.58	DIL28/SO28	20 I/O lines 8-bit timer DTMF generator		OM1071	OM1025(LCDS) + OM1028
3350A	8k	128	3.58	VSO64	30 I/O lines 8-bit timer DTMF generator 256 bytes EEPROM			
3351A	2k	64	3.58	DIL28/SO28	20 I/O lines 8-bit timer DTMF generator 128 bytes EEPROM		OM5000	
3352A	6k	128	3.58	DIL28/SO28	20 I/O lines 8-bit timer DTMF generator 128 byte EEPROM		OM5000	
3353A	6k	128	16	DIL28/SO28	20 I/O lines 8-bit timer DTMF generator Ringer out 128 bytes EEPROM	March '92	OM5000	
3354A	8k	256	16	QFP64	36 I/O lines 8-bit timer DTMF generator Ringer out 256 bytes EEPROM	June '92	OM4829 + OM5003	OM4829: Probe base
8755A	0	128	16	DIL28/SO28	8k OTP 20 I/O lines 8-bit timer DTMF generator Melody output 128 bytes EEPROM	In Development		
3301B						Piggyback for 3315, 3343, 3348	OM1083	
3344B						Piggyback for 3344, 3347, 3349	OM1071	
3346B						Piggyback for 3346	OM1076	

## CMOS and NMOS 8-bit microcontroller family

TYPE	ROM	RAM	SPEED (MHz)	PACKAGE	FUNCTIONS	REMARKS	PROBE SDS	REMARKS
3350B	OM1050	OM1050			30 I/O lines 8-bit timer DTMF generator	Piggyback for 3350A	OM4829+ OM5003	
3351B	OM1050	OM1050			30 I/O lines 8-bit timer DTMF generator	Piggyback for 3351A, 3352A, 3353A	OM5000	
3354B					30 I/O lines 8-bit timer DTMF generator	Piggyback for 3354A	OM4829+ OM5010	
3355B	OM1050 + OM1050	OM1050			30 I/O lines 8-bit timer DTMF generator			
3356B		OM1050			30 I/O lines 8-bit timer DTMF generator			
3357B	OM1050 + OM1050	OM1050 + Adapter_2			30 I/O lines 8-bit timer DTMF generator			
3358B	OM1050	OM1050			30 I/O lines 8-bit timer DTMF generator			
3359B	OM1050 + OM1050	OM1050			30 I/O lines 8-bit timer DTMF generator			
3360B					30 I/O lines 8-bit timer DTMF generator			
3361B	OM5000				30 I/O lines 8-bit timer DTMF generator			
3362B	OM5000				30 I/O lines 8-bit timer DTMF generator			
3363B	OM5000			March '82	30 I/O lines 8-bit timer DTMF generator			
3364B	OM4829 + OM5003	OM4829 + OM5003		June '82	30 I/O lines 8-bit timer DTMF generator			
3365B				In Development	30 I/O lines 8-bit timer DTMF generator			
3366B		OM1050			Piggyback for 3315, 3343, 3348			
3367B		OM1050			Piggyback for 3344, 3347, 3349			
3368B		OM1050			Piggyback for 3345			

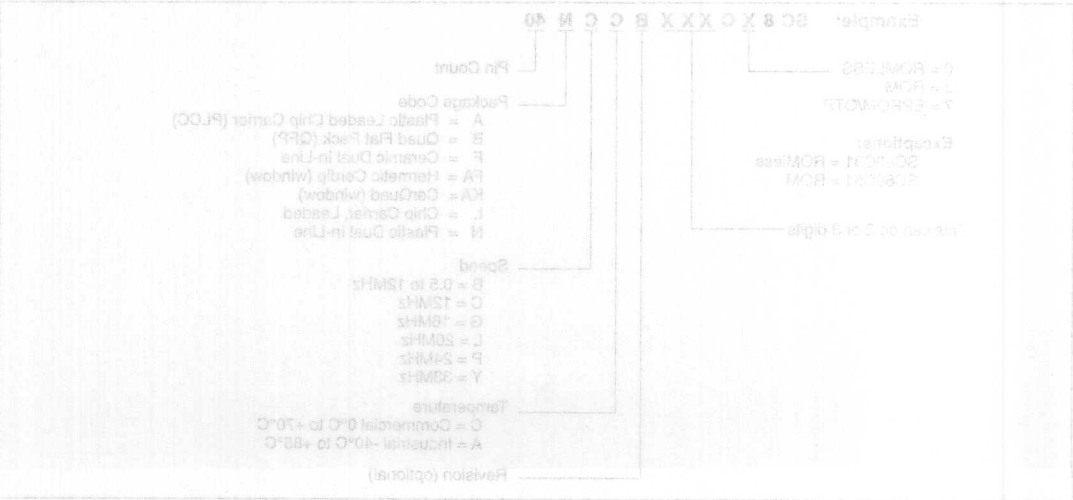
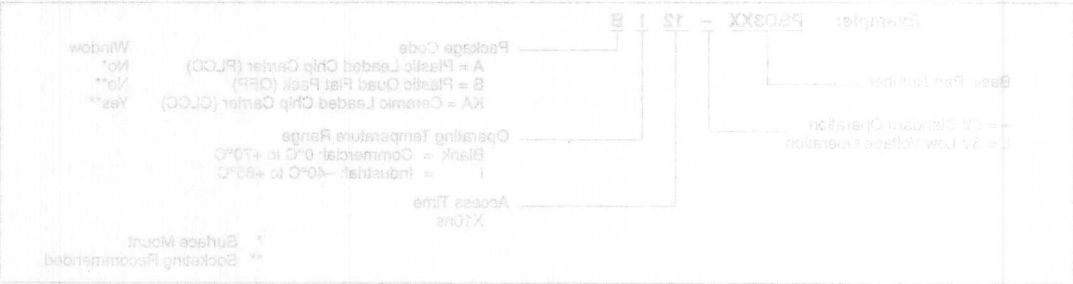
CMOS 16-bit microcontroller family

Ordering information

16-BIT CONTROLLERS

MICROCONTROLLER PRODUCTS

TYPE	(EP)ROM	RAM	SPEED (MHz)	FUNCTIONS	REMARKS	DEVELOPMENT TOOLS
68070	—	—	17.5	2 DMA channels, MMU, UART, 16-bit timer, I <sup>2</sup> C, 68000 bus interface, 16Mb address range		OM4160 Microcore OM4161 (SBE68070) TRACE32-ICE68070 (Lauterbach) OM4222 90C Development system (planned)
93C101	34k	512	15	Derivative with low power modes	Not for new design	
90C100 93C100 97C100	— 34k 32k (EPROM)	512 512 512	15 15 15	UART, I <sup>2</sup> C, 3 16-bit timers, 80C51 interface, 68000 interface, 40 I/O lines, 2Mb address range	Not for new design	OM4160/3 Microcore 3 OM4220 90C Development system TRACE32 — (Lauterbach)
90CE201	16MB external ROM	16MB external RAM	24	UART, fast I <sup>2</sup> C, 3 timers (16 bit), Watchdog timer, 68000 software compatible, EMC, QFP64	–25 to +85°C	TRACE32 — (Lauterbach)





## Ordering Information

## MICROCONTROLLER PRODUCTS

**Example:** P 8 X C X X X E B P N

0 = ROMLESS  
3 = ROM  
7 = EPROM/OTP

**Exceptions:**  
P80C32 = ROMless  
P80C52 = ROM

This can be 2 or 3 digits

**Speed**  
C = 12MHz  
E = 3.5MHz to 16MHz  
F = 1.2MHz to 16MHz  
G = 20MHz  
H = 32kHz to 12MHz  
I = 24MHz  
P = 40MHz

**FUNCTION**  
P = Philips North America Package Code  
A = Plastic Leaded Chip Carrier (PLCC)  
B = Quad Flat Pack (QFP)  
FA = Hermetic Cerdip (window)  
KA = CerQuad (window)  
N = Plastic Dual In-Line

**TEMPERATURE**  
P = Philips Package Code  
A = Plastic Leaded Chip Carrier (PLCC)  
B = Quad Flat Pack (QFP)  
F = Hermetic Cerdip (window)  
L = Cerquad (window)  
P = Plastic Dual In-Line

**TEMPERATURE**  
B = 0°C to +70°C  
F = -40°C to +85°C  
H = -40°C to +125°C

**Example:** PSD3XX - 12 I B

Basic Part Number \_\_\_\_\_

- = 5V Standard Operation  
L = 3V Low Voltage Operation

Package Code  
A = Plastic Leaded Chip Carrier (PLCC)  
B = Plastic Quad Flat Pack (QFP)  
KA = Ceramic Leaded Chip Carrier (CLCC)

Operating Temperature Range  
Blank = Commercial: 0°C to +70°C  
I = Industrial: -40°C to +85°C

Access Time  
X10ns

Window  
No\*  
No\*\*  
Yes\*\*

\* Surface Mount  
\*\* Socketing Recommended

**Example:** SC 8 X C X X X B C N 40

0 = ROMLESS  
3 = ROM  
7 = EPROM/OTP

**Exceptions:**  
SC80C31 = ROMless  
SC80C51 = ROM

This can be 2 or 3 digits

Pin Count

Package Code

- A = Plastic Leaded Chip Carrier (PLCC)
- B = Quad Flat Pack (QFP)
- F = Ceramic Dual In-Line
- FA = Hermetic Cerdip (window)
- KA = CerQuad (window)
- L = Chip Carrier, Leaded
- N = Plastic Dual In-Line

Speed

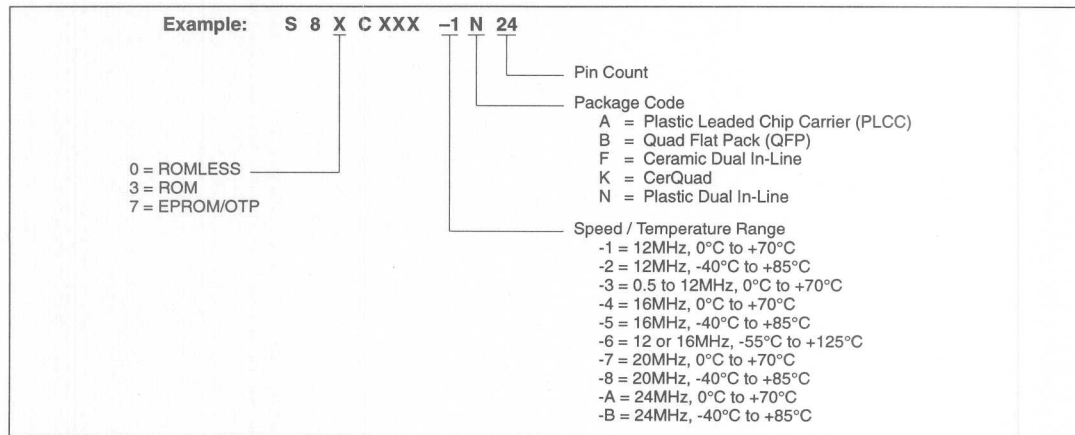
- B = 0.5 to 12MHz
- C = 12MHz
- G = 16MHz
- L = 20MHz
- P = 24MHz
- Y = 33MHz

Temperature

- C = Commercial 0°C to +70°C
- A = Industrial -40°C to +85°C

Revision (optional)

## Ordering Information





## Section 2

### 80C51 Technical Description

#### 80C51-Based 8-Bit Microcontrollers

#### CONTENTS

80C51 architecture .....	33
80C51 hardware description .....	48
80C51 programmer's guide and instruction set .....	73
80C51 EPROM products .....	128





## 80C51 Family

## 80C51 family architecture

## 80C51 ARCHITECTURE

## MEMORY ORGANIZATION

All 80C51 devices have separate address spaces for program and data memory, as shown in Figures 1 and 2. The logical separation of program and data memory allows the data memory to be accessed by 8-bit addresses, which can be quickly stored and manipulated by an 8-bit CPU. Nevertheless, 16-bit data memory addresses can also be generated through the DPTR register.

Program memory (ROM, EPROM) can only be read, not written to. There can be up to 64k bytes of program memory. In the 80C51, the lowest 4k bytes of program are on-chip. In the ROMless versions, all program memory is external. The read strobe for external program memory is the PSEN (program store enable).

Data Memory (RAM) occupies a separate address space from Program Memory. In the 80C51, the lowest 128 bytes of data memory are on-chip. Up to 64k bytes of external RAM can be addressed in the external Data Memory space. In the ROMless version, the lowest 128 bytes are on-chip. The CPU generates read and write signals, RD and WR, as needed during external Data Memory accesses.

External Program Memory and external Data Memory may be combined if desired by applying the RD and PSEN signals to the inputs of an AND gate and using the output of

the gate as the read strobe to the external Program/Data memory.

## Program Memory

Figure 3 shows a map of the lower part of the Program Memory. After reset, the CPU begins execution from location 0000H. As shown in Figure 3, each interrupt is assigned a fixed location in Program Memory. The interrupt causes the CPU to jump to that location, where it commences execution of the service routine. External Interrupt 0, for example, is assigned to location 0003H. If External Interrupt 0 is going to be used, its service routine must begin at location 0003H. If the interrupt is not going to be used, its service location is available as general purpose Program Memory.

The interrupt service locations are spaced at 8-byte intervals: 0003H for External Interrupt 0, 000BH for Timer 0, 0013H for External Interrupt 1, 001BH for Timer 1, etc. If an interrupt service routine is short enough (as is often the case in control applications), it can reside entirely within that 8-byte interval. Longer service routines can use a jump instruction to skip over subsequent interrupt locations, if other interrupts are in use.

The lowest 4k bytes of Program Memory can either be in the on-chip ROM or in an external ROM. This selection is made by strapping the EA (External Access) pin to either V<sub>CC</sub>, or V<sub>SS</sub>. In the 80C51, if the EA pin is strapped to V<sub>CC</sub>, then the program fetches to addresses 0000H through 0FFFFH are directed to the internal ROM. Program

fetches to addresses 1000H through FFFFH are directed to external ROM.

If the EA pin is strapped to V<sub>SS</sub>, then all program fetches are directed to external ROM. The ROMless parts (8031, 80C31, etc.) must have this pin externally strapped to V<sub>SS</sub> to enable them to execute from external Program Memory.

The read strobe to external ROM, PSEN, is used for all external program fetches. PSEN is not activated for internal program fetches.

The hardware configuration for external program execution is shown in Figure 4. Note that 16 I/O lines (Ports 0 and 2) are dedicated to bus functions during external Program Memory fetches. Port 0 (P0 in Figure 4) serves as a multiplexed address/data bus. It emits the low byte of the Program Counter (PCL) as an address, and then goes into a float state awaiting the arrival of the code byte from the Program Memory. During the time that the low byte of the Program Counter is valid on Port 0, the signal ALE (Address Latch Enable) clocks this byte into an address latch. Meanwhile, Port 2 (P2 in Figure 4) emits the high byte of the Program Counter (PCH). Then PSEN strobes the EPROM and the code byte is read into the microcontroller.

Program Memory addresses are always 16 bits wide, even though the actual amount of Program Memory used may be less than 64k bytes. External program execution sacrifices two of the 8-bit ports, P0 and P2, to the function of addressing the Program Memory.

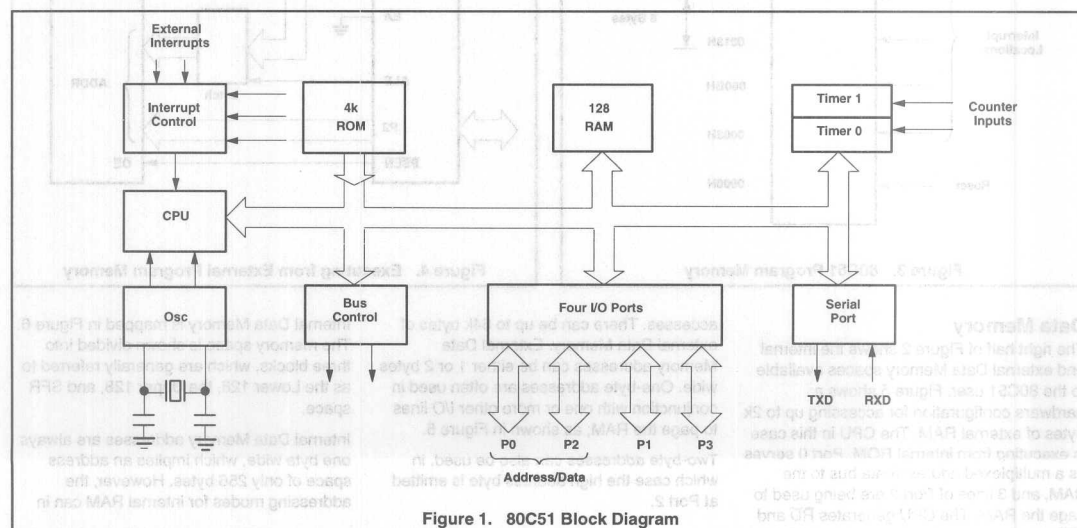
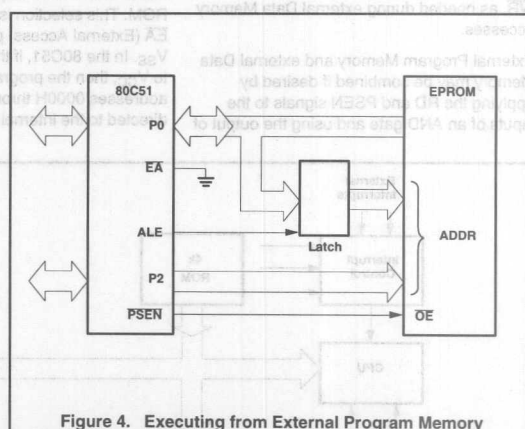
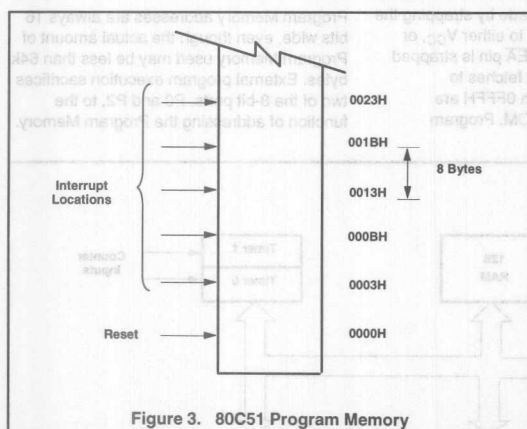
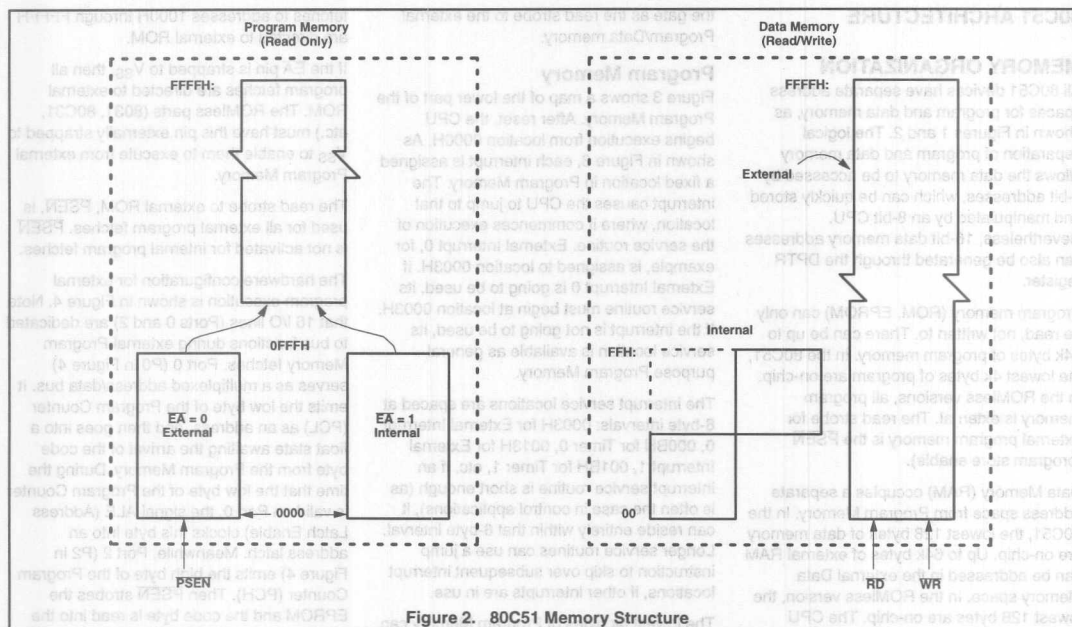


Figure 1. 80C51 Block Diagram

## 80C51 Family

## 80C51 family architecture



## Data Memory

The right half of Figure 2 shows the internal and external Data Memory spaces available to the 80C51 user. Figure 5 shows a hardware configuration for accessing up to 2k bytes of external RAM. The CPU in this case is executing from internal ROM. Port 0 serves as a multiplexed address/data bus to the RAM, and 3 lines of Port 2 are being used to page the RAM. The CPU generates RD and WR signals as needed during external RAM

accesses. There can be up to 64k bytes of external Data Memory. External Data Memory addresses can be either 1 or 2 bytes wide. One-byte addresses are often used in conjunction with one or more other I/O lines to page the RAM, as shown in Figure 5.

Two-byte addresses can also be used, in which case the high address byte is emitted at Port 2.

Internal Data Memory is mapped in Figure 6. The memory space is shown divided into three blocks, which are generally referred to as the Lower 128, the Upper 128, and SFR space.

Internal Data Memory addresses are always one byte wide, which implies an address space of only 256 bytes. However, the addressing modes for internal RAM can in

## 80C51 Family

## 80C51 family architecture

fact accommodate 384 bytes, using a simple trick. Direct addresses higher than 7FH access one memory space, and indirect addresses higher than 7FH0 access a different memory space. Thus Figure 6 shows the Upper 128 and SFR space occupying the same block of addresses, 80H through FFH, although they are physically separate entities.

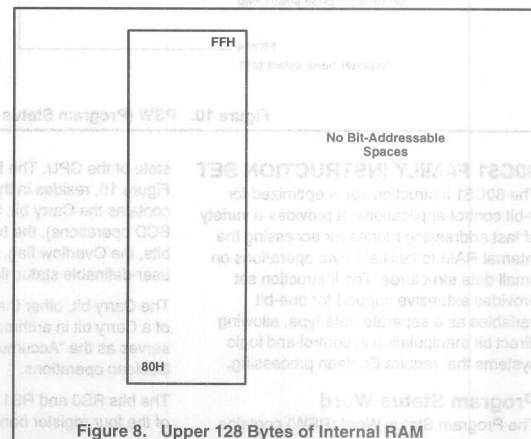
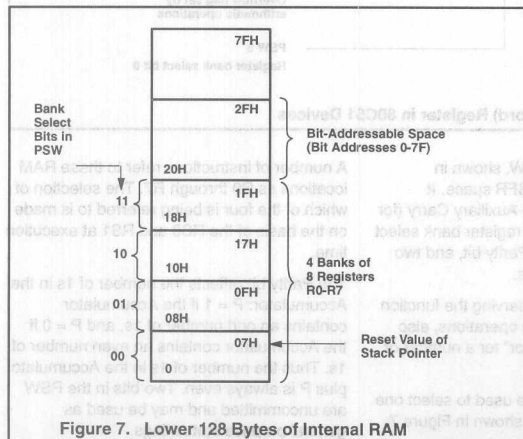
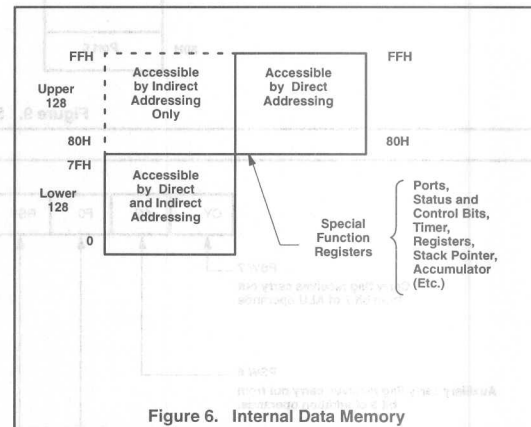
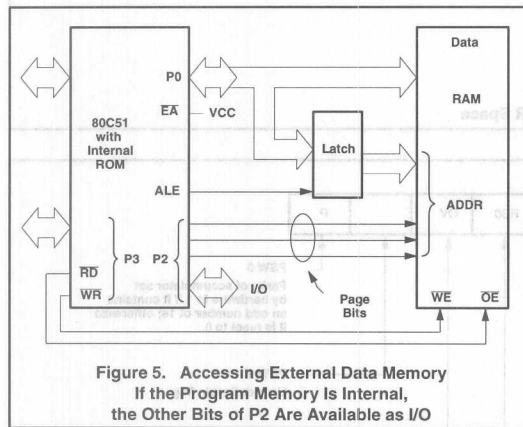
The Lower 128 bytes of RAM are present in all 80C51 devices as mapped in Figure 7. The lowest 32 bytes are grouped into 4 banks of 8 registers. Program instructions call out

these registers as R0 through R7. Two bits in the Program Status Word (PSW) select which register bank is in use. This allows more efficient use of code space, since register instructions are shorter than instructions that use direct addressing.

The next 16 bytes above the register banks form a block of bit-addressable memory space. The 80C51 instruction set includes a wide selection of single-bit instructions, and the 128 bits in this area can be directly addressed by these instructions. The bit addresses in this area are 00H through 7FH.

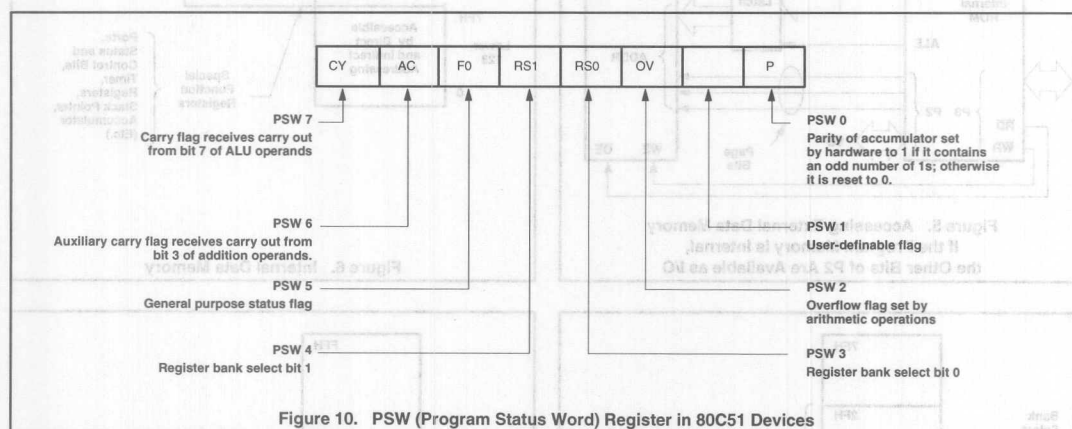
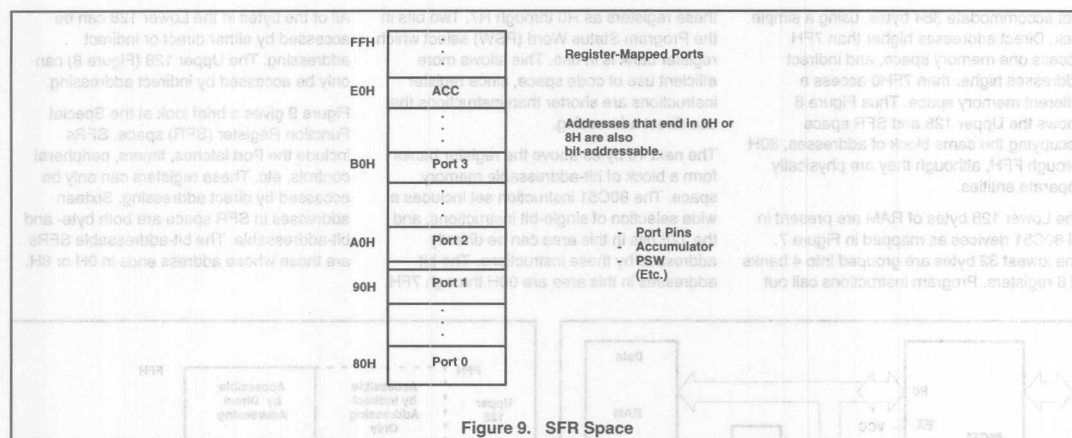
All of the bytes in the Lower 128 can be accessed by either direct or indirect addressing. The Upper 128 (Figure 8) can only be accessed by indirect addressing.

Figure 9 gives a brief look at the Special Function Register (SFR) space. SFRs include the Port latches, timers, peripheral controls, etc. These registers can only be accessed by direct addressing. Sixteen addresses in SFR space are both byte- and bit-addressable. The bit-addressable SFRs are those whose address ends in 0H or 8H.



## 80C51 Family

## 80C51 family architecture



## 80C51 FAMILY INSTRUCTION SET

The 80C51 instruction set is optimized for 8-bit control applications. It provides a variety of fast addressing modes for accessing the internal RAM to facilitate byte operations on small data structures. The instruction set provides extensive support for one-bit variables as a separate data type, allowing direct bit manipulation in control and logic systems that require Boolean processing.

## Program Status Word

The Program Status Word (PSW) contains several status bits that reflect the current

state of the CPU. The PSW, shown in Figure 10, resides in the SFR space. It contains the Carry bit, the Auxiliary Carry (for BCD operations), the two register bank select bits, the Overflow flag, a Parity bit, and two user-definable status flags.

The Carry bit, other than serving the function of a Carry bit in arithmetic operations, also serves as the "Accumulator" for a number of Boolean operations.

The bits RS0 and RS1 are used to select one of the four register banks shown in Figure 7.

A number of instructions refer to these RAM locations as R0 through R7. The selection of which of the four is being referred to is made on the basis of the RS0 and RS1 at execution time.

The Parity bit reflects the number of 1s in the Accumulator: P = 1 if the Accumulator contains an odd number of 1s, and P = 0 if the Accumulator contains an even number of 1s. Thus the number of 1s in the Accumulator plus P is always even. Two bits in the PSW are uncommitted and may be used as general purpose status flags.

## 80C51 Family

## 80C51 family architecture

**Addressing Modes**

The addressing modes in the 80C51 instruction set are as follows:

**Direct Addressing**

In direct addressing the operand is specified by an 8-bit address field in the instruction. Only internal Data RAM and SFRs can be directly addressed.

**Indirect Addressing**

In indirect addressing the instruction specifies a register which contains the address of the operand. Both internal and external RAM can be indirectly addressed.

The address register for 8-bit addresses can be R0 or R1 of the selected bank, or the Stack Pointer. The address register for 16-bit addresses can only be the 16-bit "data pointer" register, DPTR.

**Register Instructions**

The register banks, containing registers R0 through R7, can be accessed by certain instructions which carry a 3-bit register specification within the opcode of the instruction. Instructions that access the registers this way are code efficient, since this mode eliminates an address byte. When the instruction is executed, one of the eight registers in the selected bank is accessed. One of four banks is selected at execution time by the two bank select bits in the PSW.

**Register-Specific Instructions**

Some instructions are specific to a certain register. For example, some instructions always operate on the Accumulator, or Data Pointer, etc., so no address byte is needed to point to it. The opcode itself does that. Instructions that refer to the Accumulator as A assemble as accumulator specific opcodes.

**Immediate Constants**

The value of a constant can follow the opcode in Program Memory. For example,

```
MOV A, #100
```

loads the Accumulator with the decimal number 100. The same number could be specified in hex digits as 64H.

**Indexed Addressing**

Only program Memory can be accessed with indexed addressing, and it can only be read. This addressing mode is intended for reading look-up tables in Program Memory. A 16-bit base register (either DPTR or the Program Counter) points to the base of the table, and the Accumulator is set up with the table entry number.

The address of the table entry in Program Memory is formed by adding the Accumulator data to the base pointer.

Another type of indexed addressing is used in the "case jump" instruction. In this case the destination address of a jump instruction is computed as the sum of the base pointer and the Accumulator data.

**Arithmetic Instructions**

The menu of arithmetic instructions is listed in Table 1. The table indicates the addressing modes that can be used with each instruction to access the <byte> operand. For example, the ADD A,<byte> instruction can be written as:

```
ADD a, 7FH (direct addressing)
ADD A, @R0 (indirect addressing)
ADD a, R7 (register addressing)
ADD A, #127 (immediate constant)
```

The execution times listed in Table 1 assume a 12MHz clock frequency. All of the arithmetic

instructions execute in 1μs except the INC DPTR instruction, which takes 2μs, and the Multiply and Divide instructions, which take 4μs.

Note that any byte in the internal Data Memory space can be incremented without going through the Accumulator.

One of the INC instructions operates on the 16-bit Data Pointer. The Data Pointer is used to generate 16-bit addresses for external memory, so being able to increment it in one 16-bit operation is a useful feature.

The MUL AB instruction multiplies the Accumulator by the data in the B register and puts the 16-bit product into the concatenated B and Accumulator registers.

The DIV AB instruction divides the Accumulator by the data in the B register and leaves the 8-bit quotient in the Accumulator, and the 8-bit remainder in the B register.

Oddly enough, DIV AB finds less use in arithmetic "divide" routines than in radix conversions and programmable shift operations. An example of the use of DIV AB in a radix conversion will be given later. In shift operations, dividing a number by 2<sup>n</sup> shifts its n bits to the right. Using DIV AB to perform the division completes the shift in 4μs and leaves the B register holding the bits that were shifted out. The DA A instruction is for BCD arithmetic operations. In BCD arithmetic, ADD and ADDC instructions should always be followed by a DA A operation, to ensure that the result is also in BCD. Note that DA A will not convert a binary number to BCD. The DA A operation produces a meaningful result only as the second step in the addition of two BCD bytes.

**Table 1. 80C51 Arithmetic Instructions**

MNEMONIC	OPERATION	ADDRESSING MODES				EXECUTION TIME (μs)
		DIR	IND	REG	IMM	
ADD A,<byte>	A = A + <byte>	X	X	X	X	1
ADDC A,<byte>	A = A + <byte> + C	X	X	X	X	1
SUBB A,<byte>	A = A - <byte> - C	X	X	X	X	1
INC A	A = A + 1			Accumulator only		1
INC <byte>	<byte> = <byte> + 1	X	X	X		1
INC DPTR	DPTR = DPTR + 1			Data Pointer only		2
DEC A	A = A - 1			Accumulator only		1
DEC <byte>	<byte> = <byte> - 1	X	X	X		1
MUL AB	B:A = B x A			ACC and B only		4
DIV AB	A = Int[A/B] B = Mod[A/B]			ACC and B only		4
DA A	Decimal Adjust			Accumulator only		1



## 80C51 Family

## 80C51 family architecture

## Logical Instructions

Table 2 shows the list of 80C51 logical instructions. The instructions that perform Boolean operations (AND, OR, Exclusive OR, NOT) on bytes perform the operation on a bit-by-bit basis. That is, if the Accumulator contains 00110101B and byte contains 01010011B, then:

```
ANL A, <byte>
```

will leave the Accumulator holding 00010001B.

The addressing modes that can be used to access the <byte> operand are listed in Table 2.

The ANL A, <byte> instruction may take any of the forms:

```
ANL A, 7FH (direct addressing)
ANL A, @R1 (indirect addressing)
ANL A, R6 (register addressing)
ANL A, #53H (immediate constant)
```

All of the logical instructions that are Accumulator-specific execute in 1µs (using a 12MHz clock). The others take 2µs.

Note that Boolean operations can be performed on any byte in the internal Data Memory space without going through the Accumulator. The XRL <byte>, #data instruction, for example, offers a quick and easy way to invert port bits, as in XRL P1, #0FFH.

If the operation is in response to an interrupt, not using the Accumulator saves the time and effort to push it onto the stack in the service routine.

The Rotate instructions (RL, A, RLC A, etc.) shift the Accumulator 1 bit to the left or right. For a left rotation, the MSB rolls into the LSB position. For a right rotation, the LSB rolls into the MSB position.

The SWAP A instruction interchanges the high and low nibbles within the Accumulator. This is a useful operation in BCD manipulations. For example, if the Accumulator contains a binary number which is known to be less than 100, it can be quickly converted to BCD by the following code:

```
MOVE B, #10
DIV AB
SWAP A
ADD A, B
```

Dividing the number by 10 leaves the tens digit in the low nibble of the Accumulator, and the ones digit in the B register. The SWAP and ADD instructions move the tens digit to the high nibble of the Accumulator, and the ones digit to the low nibble.

## Data Transfers

## Internal RAM

Table 3 shows the menu of instructions that are available for moving data around within

the internal memory spaces, and the addressing modes that can be used with each one. With a 12MHz clock, all of these instructions execute in either 1 or 2µs.

The MOV <dest>, <src> instruction allows data to be transferred between any two internal RAM or SFR locations without going through the Accumulator. Remember, the Upper 128 bytes of data RAM can be accessed only by indirect addressing, and SFR space only by direct addressing.

Note that in 80C51 devices, the stack resides in on-chip RAM, and grows upwards. The PUSH instruction first increments the Stack Pointer (SP), then copies the byte into the stack. PUSH and POP use only direct addressing to identify the byte being saved or restored, but the stack itself is accessed by indirect addressing using the SP register. This means the stack can go into the Upper 128 bytes of RAM, if they are implemented, but not into SFR space.

The Upper 128 bytes of RAM are not implemented in the 80C51 nor in its ROMless or EPROM counterparts. With these devices, if the SP points to the Upper 128, PUSHed bytes are lost, and POPed bytes are indeterminate.

The Data Transfer instructions include a 16-bit MOV that can be used to initialize the Data Pointer (DPTR) for look-up tables in Program Memory, or for 16-bit external Data Memory accesses.

Table 2. 80C51 Logical Instructions

MNEMONIC	OPERATION	ADDRESSING MODES				EXECUTION TIME (µs)
		DIR	IND	REG	IMM	
ANL A, <byte>	A = A.AND. <byte>	X	X	X	X	1
ANL <byte>, A	<byte> = <byte> .AND.A	X				1
ANL <byte>, #data	<byte> = <byte> .AND.#data	X				2
ORL A, <byte>	A = A.OR. <byte>	X	X	X	X	1
ORL <byte>, A	<byte> = <byte> .OR.A	X				1
ORL <byte>, #data	<byte> = <byte> .OR.#data	X				2
XRL A, <byte>	A = A.XOR. <byte>	X	X	X	X	1
XRL <byte>, A	<byte> = <byte> .XOR.A	X				1
XRL <byte>, #data	<byte> = <byte> .XOR.#data	X				2
CPL A	A = 00H					1
CPL A	A = .NOT.A					1
RL A	Rotate ACC Left 1 bit					1
RLC A	Rotate Left through Carry					1
RR A	Rotate ACC Right 1 bit					1
RRC A	Rotate Right through Carry					1
SWAP A	Swap Nibbles in A					1

## 80C51 Family

## 80C51 family architecture

Table 3. Data Transfer Instructions that Access Internal Data Memory Space

MNEMONIC (8-bit)	OPERATION	ADDRESSING MODES				EXECUTION TIME (μs)
		DIR	IND	REG	IMM	
MOV A,<src>	A = <src>	X	X	X	X	1
MOV <dest>,A	<dest> = A	X	X	X		1
MOV <dest>,<src>	<dest> = <src>	X	X	X	X	2
MOV DPTR,#data16	DPTR = 16-bit immediate constant				X	2
PUSH <src>	INC SP:MOV@SP,<src>	X				2
POP <dest>	MOV <dest>,@SP:DEC SP	X				2
XCH A,<byte>	ACC and <byte> exchange data	X	X	X		1
XCHD A,@Ri	ACC and @Ri exchange low nibbles		X			1

The XCH A, <byte> instruction causes the Accumulator and addressed byte to exchange data. The XCHD A, @Ri instruction is similar, but only the low nibbles are involved in the exchange.

To see how XCH and XCHD can be used to facilitate data manipulations, consider first the problem of shifting an 8-digit BCD number two digits to the right. Figure 11 shows how this can be done using direct MOVs, and for comparison how it can be done using XCH instructions. To aid in understanding how the code works, the contents of the registers that are holding the BCD number and the content of the Accumulator are shown alongside each instruction to indicate their status after the instruction has been executed.

After the routine has been executed, the Accumulator contains the two digits that were shifted out on the right. Doing the routine with direct MOVs uses 14 code bytes and 9 μs of execution time (assuming a 12MHz clock). The same operation with XCHs uses only 9 bytes and executes almost twice as fast.

	2A	2B	2C	2D	2E	ACC
MOV A,2EH	00	12	34	56	78	78
MOV 2EH,2DH	00	12	34	56	56	78
MOV 2DH,2CH	00	12	34	34	56	78
MOV 2CH,2BH	00	12	12	34	56	78
MOV 2BH,#0	00	00	12	34	56	78
A. Using direct MOVs: 14 bytes, 9 μs						
	2A	2B	2C	2D	2E	ACC
CLR A	00	12	34	56	78	00
XCH A,2BH	00	00	34	56	78	12
XCH A,2CH	00	00	12	56	78	34
XCH A,2DH	00	00	12	34	78	56
XCH A,2EH	00	00	12	34	56	78
B. Using XCHs: 9 bytes, 5 μs						

Figure 11. Shifting a BCD Number Two Digits to the Right

To right-shift by an odd number of digits, a one-digit shift must be executed.

Figure 12 shows a sample of code that will right-shift a BCD number one digit, using the XCHD instruction. Again, the contents of the registers holding the number and of the Accumulator are shown alongside each instruction.

First, pointers R1 and R0 are set up to point to the two bytes containing the last four BCD digits. Then a loop is executed which leaves the last byte, location 2EH, holding the last two digits of the shifted number. The pointers are decremented, and the loop is repeated for location 2DH. The CJNE instruction (Compare and Jump if Not Equal) is a loop control that will be described later. The loop executed from LOOP to CJNE for R1 = 2EH, 2DH, 2CH, and 2BH. At that point the digit that was originally shifted out on the right has propagated to location 2AH. Since that location should be left with 0s, the last digit is moved to the Accumulator.

	2A	2B	2C	2D	2E	ACC
MOV R1,#2EH	00	12	34	56	78	XX
MOV R0,2DH	00	12	34	56	78	XX
loop for R1 = 2EH:						
LOOP: MOV A,@R1	00	12	34	56	78	78
XCHD A,@R0	00	12	34	58	78	76
SWAP A	00	12	34	58	78	67
MOV @R1,A	00	12	34	58	67	67
DEC R1	00	12	34	58	67	67
DEC R0	00	12	34	58	67	67
CJNE R1,#2AH,LOOP						
loop for R1 = 2DH:						
	00	12	38	45	67	45
loop for R1 = 2CH:	00	18	23	45	67	23
loop for R1 = 2BH:	08	01	23	45	67	01
loop for R1 = 2AH:						
CLR A	08	01	23	45	67	00
XCH A,2AH	00	01	23	45	67	08

Figure 12. Shifting a BCD Number One Digit to the Right

## 80C51 Family

## 80C51 family architecture

Table 4. 80C51 Data Transfer Instructions that Access External Data Memory Space

ADDRESS WIDTH	MNEMONIC	OPERATION	EXECUTION TIME (μs)
8 bits	MOVX A, @Ri	Read external RAM @ Ri	2
8 bits	MOVX @Ri, A	Write external RAM @ Ri	2
16 bits	MOVX A, @DPTR	Read external RAM @ DPTR	2
16 bits	MOVX @DPTR, A	Write external RAM @ DPTR	2

## Lookup Tables

Table 5 shows the two instructions that are available for reading lookup tables in Program Memory. Since these instructions access only Program Memory, the lookup tables can only be read, not updated.

If the table access is to external Program Memory, then the read strobe is PSEN.

The mnemonic is MOVC for "move constant." The first MOVC instruction in Table 5 can accommodate a table of up to 256 entries numbered 0 through 255. The number of the desired entry is loaded into the Accumulator, and the Data Pointer is set up to point to the beginning of the table. Then:

```
MOVC A, @A+DPTR
```

copies the desired table entry into the Accumulator.

The other MOVC instruction works the same way, except the Program Counter (PC) is used as the table base, and the table is accessed through a subroutine. First the number of the desired entry is loaded into the Accumulator, and the subroutine is called:

```
MOV A, ENTRY NUMBER
CALL TABLE
```

The subroutine "TABLE" would look like this:

```
TABLE: MOVC A, @A+PC
RET
```

The table itself immediately follows the RET (return) instruction in Program Memory. This type of table can have up to 255 entries, numbered 1 through 255. Number 0 cannot be used, because at the time the MOVC instruction is executed, the PC contains the address of the RET instruction. An entry numbered 0 would be the RET opcode itself.

## Boolean Instructions

80C51 devices contain a complete Boolean (single-bit) processor. The internal RAM

contains 128 addressable bits, and the SFR space can support up to 128 addressable bits as well. All of the port lines are

bit-addressable, and each one can be treated as a separate single-bit port. The instructions that access these bits are not just conditional branches, but a complete menu of move, set, clear, complement, OR, and AND instructions. These kinds of bit operations are not easily obtained in other architectures with any amount of byte-oriented software.

The instruction set for the Boolean processor is shown in Table 6. All bit accesses are by direct addressing.

Bit addresses 00H through 7FH are in the Lower 128, and bit addresses 80H through FFH are in SFR space.

Note how easily an internal flag can be moved to a port pin:

```
MOV C, FLAG
MOV P1.0, C
```

In this example, FLAG is the name of any addressable bit in the Lower 128 or SFR space. An I/O line (the LSB of Port 1, in this case) is set or cleared depending on whether the flag bit is 1 or 0.

The Carry bit in the PSW is used as the single-bit Accumulator of the Boolean processor. Bit instructions that refer to the Carry bit as C assemble as Carry-specific instructions (CLR C, etc.). The Carry bit also has a direct address, since it resides in the PSW register, which is bit-addressable.

Note that the Boolean instruction set includes ANL and ORL operations, but not the XRL (Exclusive OR) operation. An XRL operation is simple to implement in software. Suppose, for example, it is required to form the Exclusive OR of two bits:

C = bit1 .XRL. bit2

The software to do that could be as follows:

```
MOV C, bit1
JNB bit2, OVER
CPL C
```

OVER: (continue)

First, bit1 is moved to the Carry. If bit2 = 0, then C now contains the correct result. That is, bit1 .XRL. bit2 = bit1 if bit2 = 0. On the other hand, if bit2 = 1, C now contains the complement of the correct result. It need only be inverted (CPL C) to complete the operation.

This code uses the JNB instruction, one of a series of bit-test instructions which execute a jump if the addressed bit is set (JC, JB, JBC) or if the addressed bit is not set (JNC, JNB). In the above case, bit2 is being tested, and if bit2 = 0, the CPL C instruction is jumped over.

JBC executes the jump if the addressed bit is set, and also clears the bit. Thus a flag can be tested and cleared in one operation. All the PSW bits are directly addressable, so the Parity bit, or the general purpose flags, for example, are also available to the bit-test instructions.

## Relative Offset

The destination address for these jumps is specified to the assembler by a label or by an actual address in Program memory. However, the destination address assembles to a relative offset byte. This is a signed (two's complement) offset byte which is added to the PC in two's complement arithmetic if the jump is executed. The range of the jump is therefore -128 to +127 Program Memory bytes relative to the first byte following the instruction.

Table 5. 80C51 Lookup Table Read Instructions

MNEMONIC	OPERATION	EXECUTION TIME (μs)
MOVC A, @A+DPTR	Read program memory at (A + DPTR)	2
MOVC A, @A+PC	Read program memory at (A + PC)	2

## 80C51 Family

## 80C51 family architecture

Table 6. 80C51 Boolean Instructions

MNEMONIC	OPERATION	EXECUTION TIME ( $\mu$ s)
ANL C,bit	C = C.AND.bit	2
ANL C,/bit	C = C.AND..NOT.bit	2
ORL C,bit	C = C.OR.bit	2
ORL C,/bit	C = C.OR..NOT.bit	2
MOV C,bit	C = bit	1
MOV bit,C	bit = C	2
CLR C	C = 0	1
CLR bit	bit = 0	1
SETB C	C = 1	1
SETB bit	bit = 1	1
CPL C	C = .NOT.C	1
CPL bit	bit = .NOT.bit	1
JC rel	Jump if C = 1	2
JNC rel	Jump if C = 0	2
JB bit,rel	Jump if bit = 1	2
JNB bit,rel	Jump if bit = 0	2
JBC bit,rel	Jump if bit = 1; CLR bit	2

## Jump Instructions

Table 7 shows the list of unconditional jumps with execution time for a 12MHz clock.

The table lists a single "JMP addr" instruction, but in fact there are three SJMP, LJMP, and AJMP, which differ in the format of the destination address. JMP is a generic mnemonic which can be used if the programmer does not care which way the jump is encoded.

The SJMP instruction encodes the destination address as a relative offset, as described above. The instruction is 2 bytes long, consisting of the opcode and the relative offset byte. The jump distance is limited to a range of -128 to +127 bytes relative to the instruction following the SJMP.

The LJMP instruction encodes the destination address as a 16-bit constant. The instruction

is 3 bytes long, consisting of the opcode and two address bytes. The destination address can be anywhere in the 64k Program Memory space.

The AJMP instruction encodes the destination address as an 11-bit constant. The instruction is 2 bytes long, consisting of the opcode, which itself contains 3 of the 11 address bits, followed by another byte containing the low 8 bits of the destination address. When the instruction is executed, these 11 bits are simply substituted for the low 11 bits in the PC. The high 5 bits stay the same. Hence the destination has to be within the same 2k block as the instruction following the AJMP.

In all cases the programmer specifies the destination address to the assembler in the same way: as a label or as a 16-bit constant. The assembler will put the destination

address into the correct format for the given instruction. If the format required by the instruction will not support the distance to the specified destination address, a "Destination out of range" message is written into the List file.

The JMP @A+DPTR instruction supports case jumps. The destination address is computed at execution time as the sum of the 16-bit DPTR register and the Accumulator. Typically, DPTR is set up with the address of a jump table. In a 5-way branch, for example, an integer 0 through 4 is loaded into the Accumulator. The code to be executed might be as follows:

```
MOV DPTR,#JUMP_TABLE
MOV A,INDEX_NUMBER
RL A
JMP @A+DPTR
```

Table 7. Unconditional Jumps in 80C51 Devices

MNEMONIC	OPERATION	EXECUTION TIME ( $\mu$ s)
JMP addr	Jump to addr	2
JMP @A+DPTR	Jump to A + DPTR	2
CALL addr	Call subroutine at addr	2
RET	Return from subroutine	2
RETI	Return from interrupt	2
NOP	No operation	1



# 80C51 Family

# 80C51 family architecture

The RL A instruction converts the index number (0 through 4) to an even number on the range 0 through 8, because each entry in the jump table is 2 bytes long:

JUMP TABLE:

AJMP	CASE 0
AJMP	CASE 1
AJMP	CASE 2
AJMP	CASE 3
AJMP	CASE 4

Table 7 shows a single "CALL addr" instruction, but there are two of them, LCALL and ACALL, which differ in the format in which the subroutine address is given to the CPU. CALL is a generic mnemonic which can be used if the programmer does not care which way the address is encoded.

The LCALL instruction uses the 16-bit address format, and the subroutine can be anywhere in the 64k Program Memory space. The ACALL instruction uses the 11-bit format, and the subroutine must be in the same 2k block as the instruction following the ACALL.

In any case, the programmer specifies the subroutine address to the assembler in the same way: as a label or as a 16-bit constant. The assembler will put the address into the correct format for the given instructions.

Subroutines should end with a RET instruction, which returns execution to the instruction following the CALL.

RETI is used to return from an interrupt service routine. The only difference between RET and RETI is that RETI tells the interrupt control system that the interrupt in progress is done. If there is no interrupt in progress at the time RETI is executed, then the RETI is functionally identical to RET.

Table 8 shows the list of conditional jumps available to the 80C51 user. All of these jumps specify the destination address by the relative offset method, and so are limited to a jump distance of -128 to +127 bytes from the instruction following the conditional jump instruction. Important to note, however, the user specifies to the assembler the actual

destination address the same way as the other jumps: as a label or a 16-bit constant.

There is no Zero bit in the PSW. The JZ and JNZ instructions test the Accumulator data for that condition.

The DJNZ instruction (Decrement and Jump if Not Zero) is for loop control. To execute a loop N times, load a counter byte with N and terminate the loop with a DJNZ to the beginning of the loop, as shown below for N = 10.

```
MOV     COUNTER,#10
LOOP: (begin loop)
    •
    •
    •
    (end loop)
    DJNZ COUNTER,LOOP
    (continue)
```

The CJNE instruction (Compare and Jump if Not Equal) can also be used for loop control as in Figure 12. Two bytes are specified in the operand field of the instruction. The jump is executed only if the two bytes are not equal. In the example of Figure 12, the two bytes were data in R1 and the constant 2AH. The initial data in R1 was 2EH. Every time the loop was executed, R1 was decremented, and the looping was to continue until the R1 data reached 2AH.

Another application of this instruction is in "greater than, less than" comparisons. The two bytes in the operand field are taken as unsigned integers. If the first is less than the second, then the Carry bit is set (1). If the first is greater than or equal to the second, then the Carry bit is cleared.

## CPU Timing

All 80C51 microcontrollers have an on-chip oscillator which can be used if desired as the clock source for the CPU. To use the on-chip oscillator, connect a crystal or ceramic resonator between the XTAL1 and XTAL2 pins of the microcontroller, and capacitors to ground as shown in Figure 13.

Examples of how to drive the clock with an external oscillator are shown in Figure 14. Note that in the NMOS devices (8051, etc.) the signal at the XTAL2 pin actually drives the internal clock generator. In the CMOS devices (80C51, etc.), the signal at the XTAL1 pin drives the internal clock generator. The internal clock generator defines the sequence of states that make up the 80C51 machine cycle.

## Machine Cycles

A machine cycle consists of a sequence of 6 states, numbered S1 through S6. Each state time lasts for two oscillator periods. Thus a machine cycle takes 12 oscillator periods or 1µs if the oscillator frequency is 12MHz.

Each state is divided into a Phase 1 half and a Phase 2 half. Figure 15 shows that fetch/execute sequences in states and phases for various kinds of instructions. Normally two program fetches are generated during each machine cycle, even if the instruction being executed doesn't require it. If the instruction being executed doesn't need more code bytes, the CPU simply ignores the extra fetch, and the Program Counter is not incremented.

Execution of a one-cycle instruction (Figures 15a and 15b) begins during State 1 of the machine cycle, when the opcode is latched into the Instruction Register. A second fetch occurs during S4 of the same machine cycle. Execution is complete at the end of State 6 of this machine cycle.

The MOVX instructions take two machine cycles to execute. No program fetch is generated during the second cycle of a MOVX instruction. This is the only time program fetches are skipped. The fetch/execute sequence for MOVX instructions is shown in Figure 15d.

The fetch/execute sequences are the same whether the Program Memory is internal or external to the chip. Execution times do not depend on whether the Program Memory is internal or external.

Table 8. Conditional Jumps in 80C51 Devices

MNEMONIC	OPERATION	ADDRESSING MODES				EXECUTION TIME (µs)
		DIR	IND	REG	IMM	
JZ rel	Jump if A = 0			Accumulator only		2
JNZ rel	Jump if A ≠ 0			Accumulator only		2
DJNZ <byte>,rel	Decrement and jump if not zero	X		X		2
CJNE A,<byte>,rel	Jump if A ≠ <byte>	X			X	2
CJNE <byte>,#data,rel	Jump if <byte> ≠ #data		X	X		2

## 80C51 Family

## 80C51 family architecture

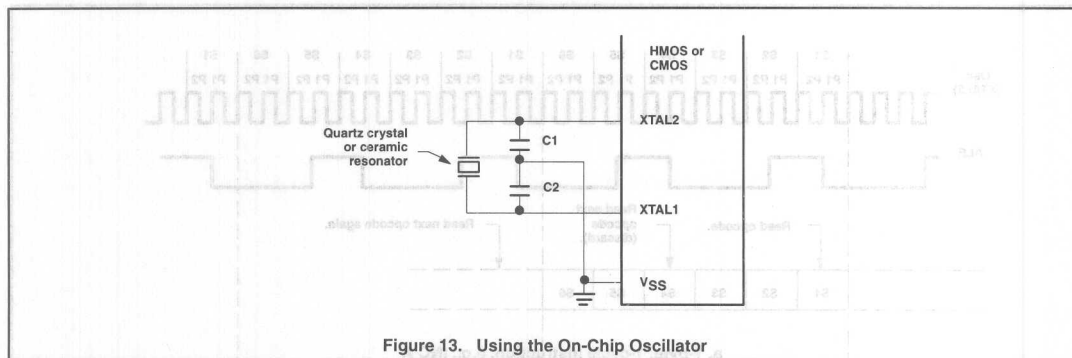


Figure 13. Using the On-Chip Oscillator

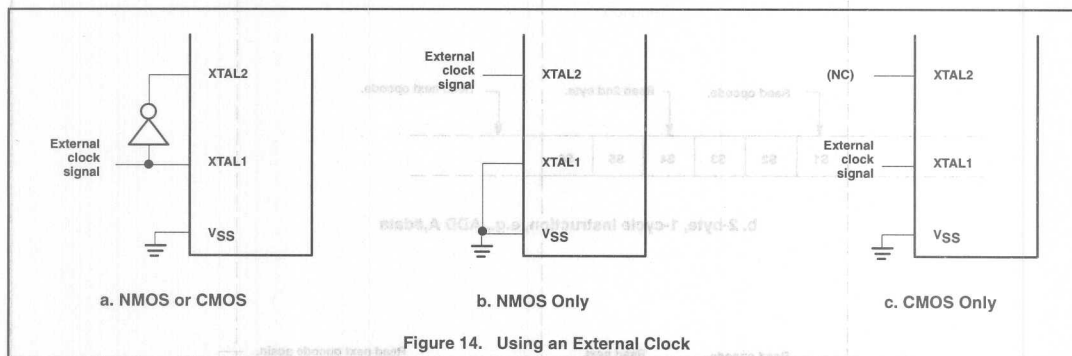


Figure 14. Using an External Clock

Figure 16 shows the signals and timing involved in program fetches when the Program Memory is external. If Program Memory is external, then the Program Memory read strobe  $\overline{\text{PSEN}}$  is normally activated twice per machine cycle, as shown in Figure 16a. If an access to external Data Memory occurs, as shown in Figure 16b, two  $\overline{\text{PSEN}}$ s are skipped, because the address and data bus are being used for the Data Memory access.

Note that a Data Memory bus cycle takes twice as much time as a Program Memory bus cycle. Figure 16 shows the relative timing of the addresses being emitted at Ports 0 and 2, and of ALE and  $\overline{\text{PSEN}}$ . ALE is used to

latch the low address byte from P0 into the address latch.

When the CPU is executing from internal Program Memory,  $\overline{\text{PSEN}}$  is not activated, and program addresses are not emitted. However, ALE continues to be activated twice per machine cycle and so it is available as a clock output signal. Note, however, that one ALE is skipped during the execution of the MOVX instruction.

### Interrupt Structure

The 80C51 and its ROMless and EPROM versions have 5 interrupt sources: 2 external interrupts, 2 timer interrupts, and the serial port interrupt.

What follows is an overview of the interrupt structure for the device. More detailed information for specific members of the 80C51 derivative family is provided in later chapters of this user's guide.

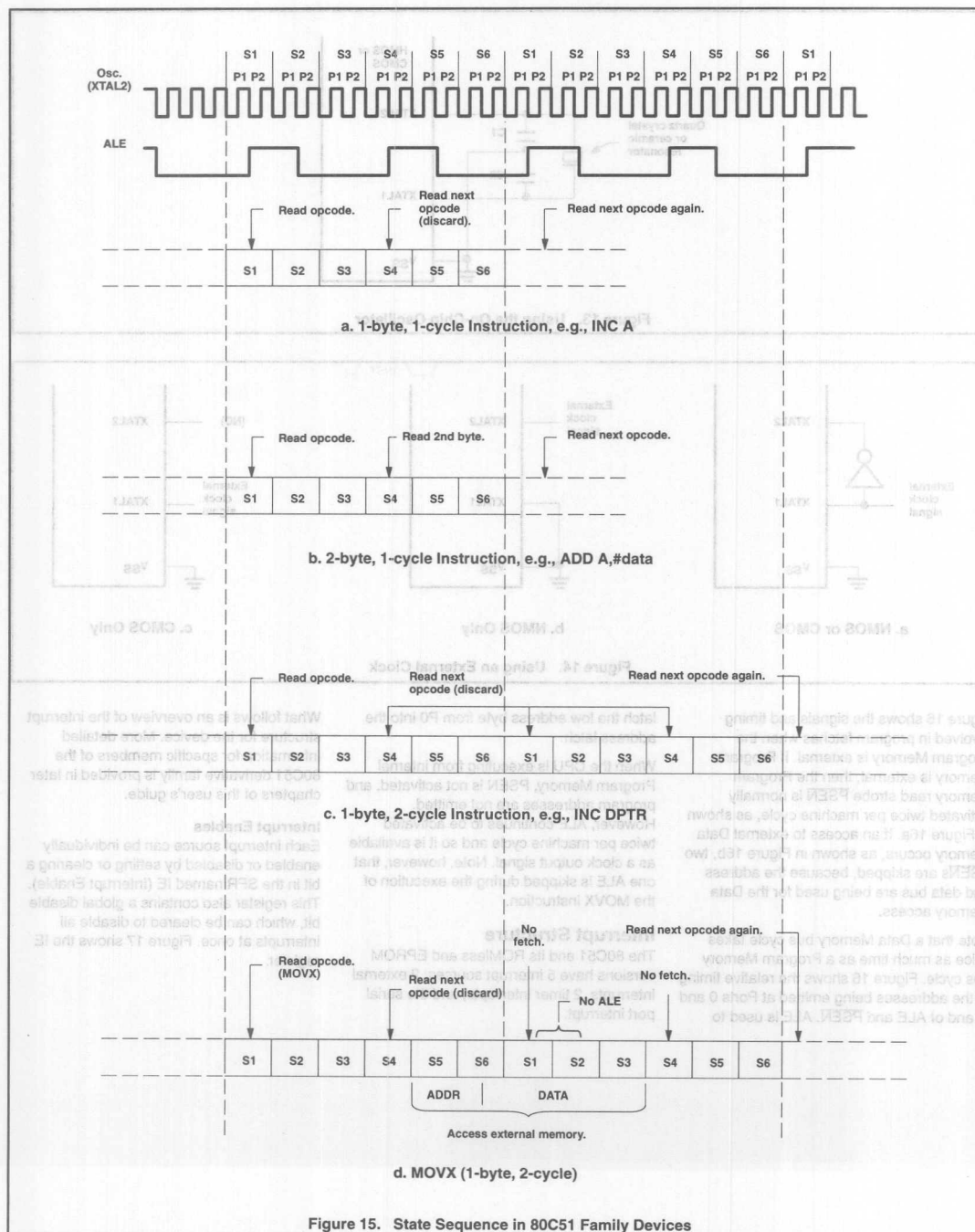
### Interrupt Enables

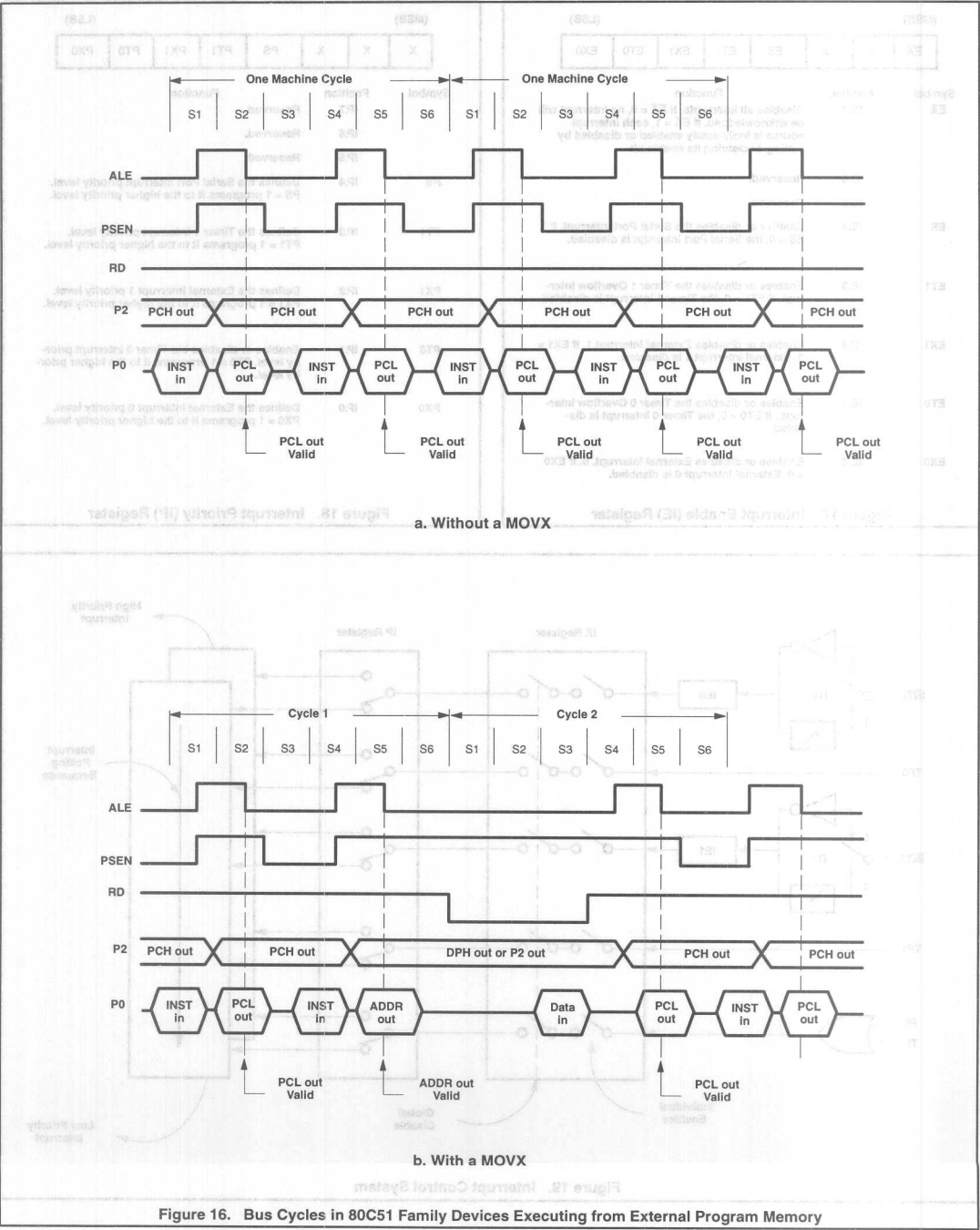
Each interrupt source can be individually enabled or disabled by setting or clearing a bit in the SFR named IE (Interrupt Enable). This register also contains a global disable bit, which can be cleared to disable all interrupts at once. Figure 17 shows the IE register.



## 80C51 Family

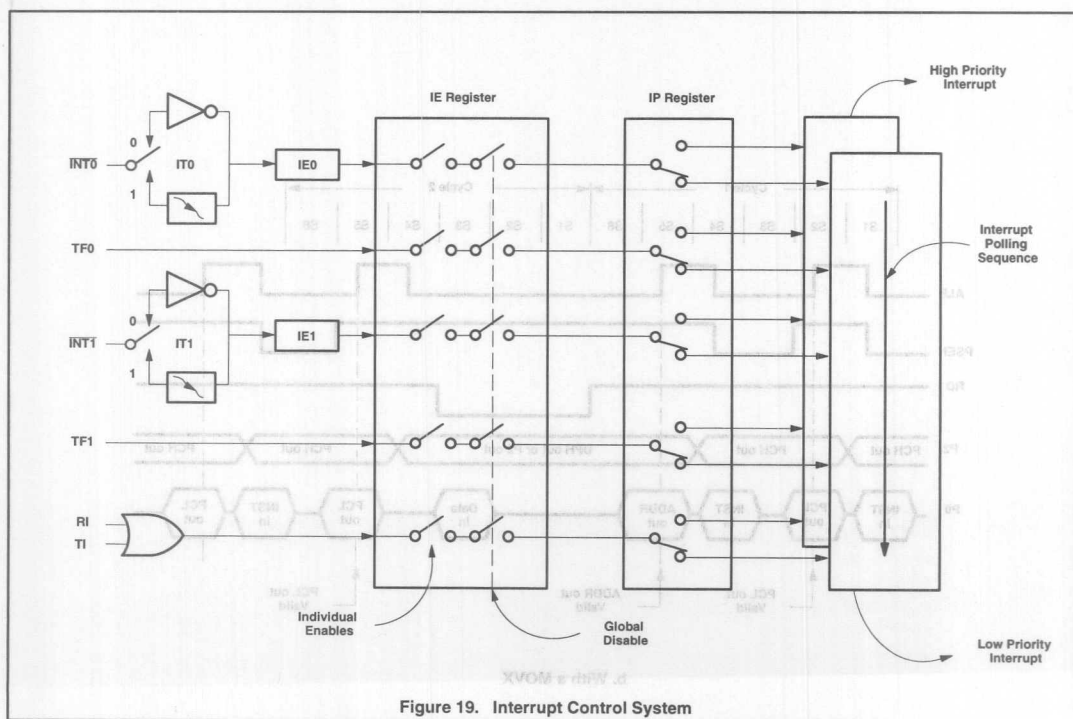
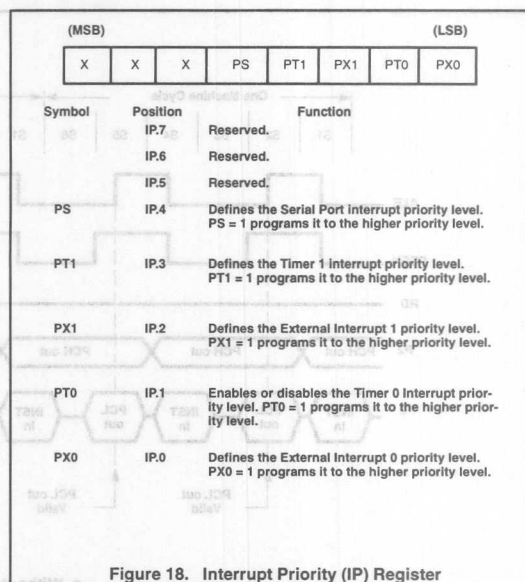
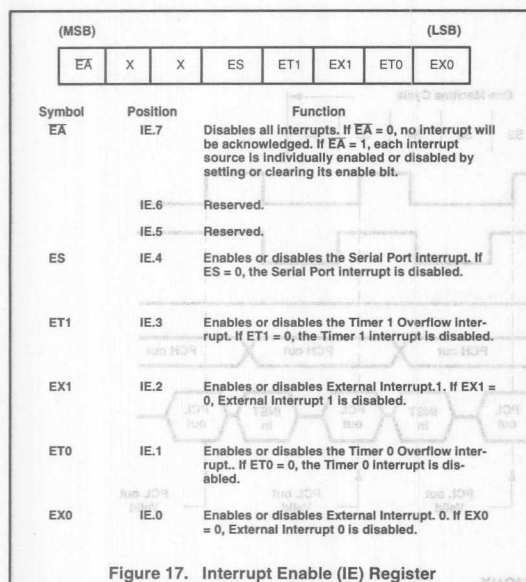
## 80C51 family architecture





## 80C51 Family

## 80C51 family architecture



## 80C51 Family

## 80C51 family architecture

**Interrupt Priorities**

Each interrupt source can also be individually programmed to one of two priority levels by setting or clearing a bit in the SFR named IP (Interrupt Priority). Figure 18 shows the IP register. A low-priority interrupt can be interrupted by a high-priority interrupt, but not by another low-priority interrupt. A high-priority interrupt can't be interrupted by any other interrupt source.

If two interrupt requests of different priority levels are received simultaneously, the request of higher priority is serviced. If interrupt requests of the same priority level are received simultaneously, an internal polling sequence determines which request is serviced. Thus within each priority level there is a second priority structure determined by the polling sequence. Figure 19 shows how the IE and IP registers and the polling sequence work to determine which if any interrupt will be serviced.

In operation, all the interrupt flags are latched into the interrupt control system during State 5 of every machine cycle. The samples are polled during the following machine cycle. If the flag for an enabled interrupt is found to be set (1), the interrupt system generates an LCALL to the appropriate location in Program Memory, unless some other condition blocks the interrupt. Several conditions can block an interrupt, among them that an interrupt of

equal or higher priority level is already in progress.

The hardware-generated LCALL causes the contents of the Program Counter to be pushed into the stack, and reloads the PC with the beginning address of the service routine. As previously noted (Figure 3), the service routine for each interrupt begins at a fixed location.

Only the Program Counter is automatically pushed onto the stack, not the PSW or any other register. Having only the PC automatically saved allows the programmer to decide how much time should be spent saving other registers. This enhances the interrupt response time, albeit at the expense of increasing the programmer's burden of responsibility. As a result, many interrupt functions that are typical in control applications toggling a port pin for example, or reloading a timer, or unloading a serial buffer can often be completed in less time than it takes other architectures to complete.

**Simulating a Third Priority Level in Software**

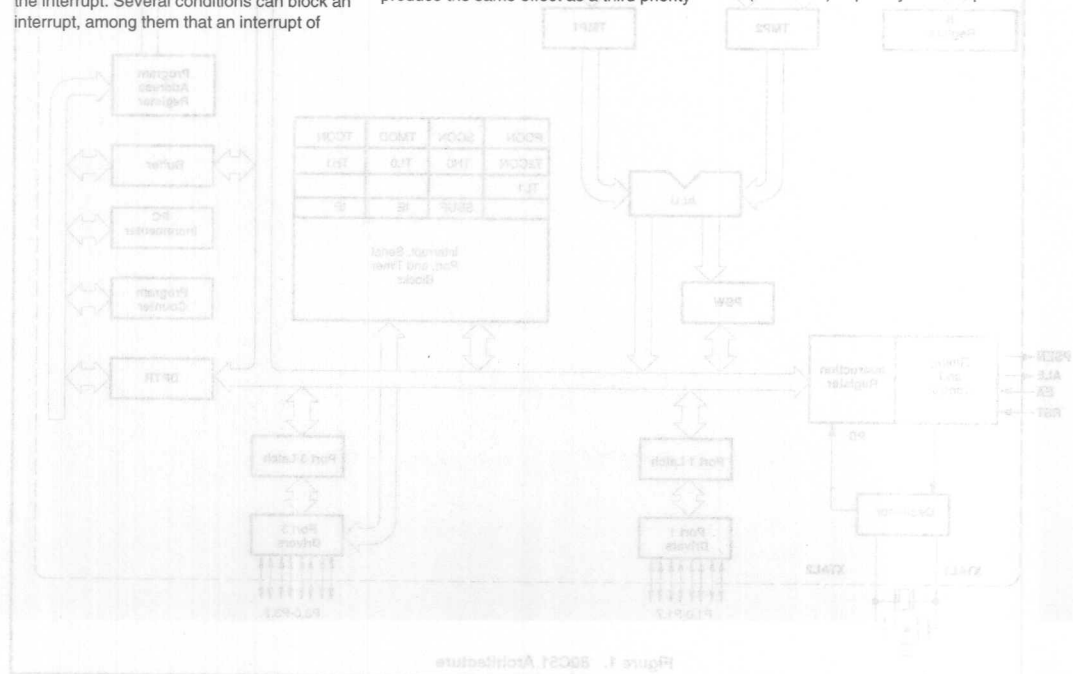
Some applications require more than two priority levels that are provided by on-chip hardware in 80C51 devices. In these cases, relatively simple software can be written to produce the same effect as a third priority

level. First, interrupts that are to have higher priority than 1 are assigned to priority 1 in the Interrupt Priority (IP) register. The service routines for priority 1 interrupts that are supposed to be interruptable by priority 2 interrupts are written to include the following code:

```
PUSH IE
MOV IE,#MASK
CALL LABEL
*****
(execute service routine)
*****
POP IE
RET
LABEL: RETI
```

As soon as any priority interrupt is acknowledged, the Interrupt Enable (IE) register is redefined so as to disable all but priority 2 interrupts. Then a CALL to LABEL executes the RETI instruction, which clears the priority 1 interrupt-in-progress flip-flop. At this point any priority 1 interrupt that is enabled can be serviced, but only priority 2 interrupts are enabled.

POPing IE restores the original enable byte. Then a normal RET (rather than another RETI) is used to terminate the service routine. The additional software adds 10µs (at 12MHz) to priority 1 interrupts.



## 80C51 Family

## 80C51 family hardware description

## HARDWARE DESCRIPTION

This chapter provides a detailed description of the 80C51 microcontroller (see Figure 1). Included in this description are:

- The port drivers and how they function both as ports and, for Ports 0 and 2, in bus operations

- The Timers/Counters

- The Serial Interface

- The Interrupt System

- Reset

- The Reduced Power Modes in CMOS devices

- The EPROM version of the 80C51

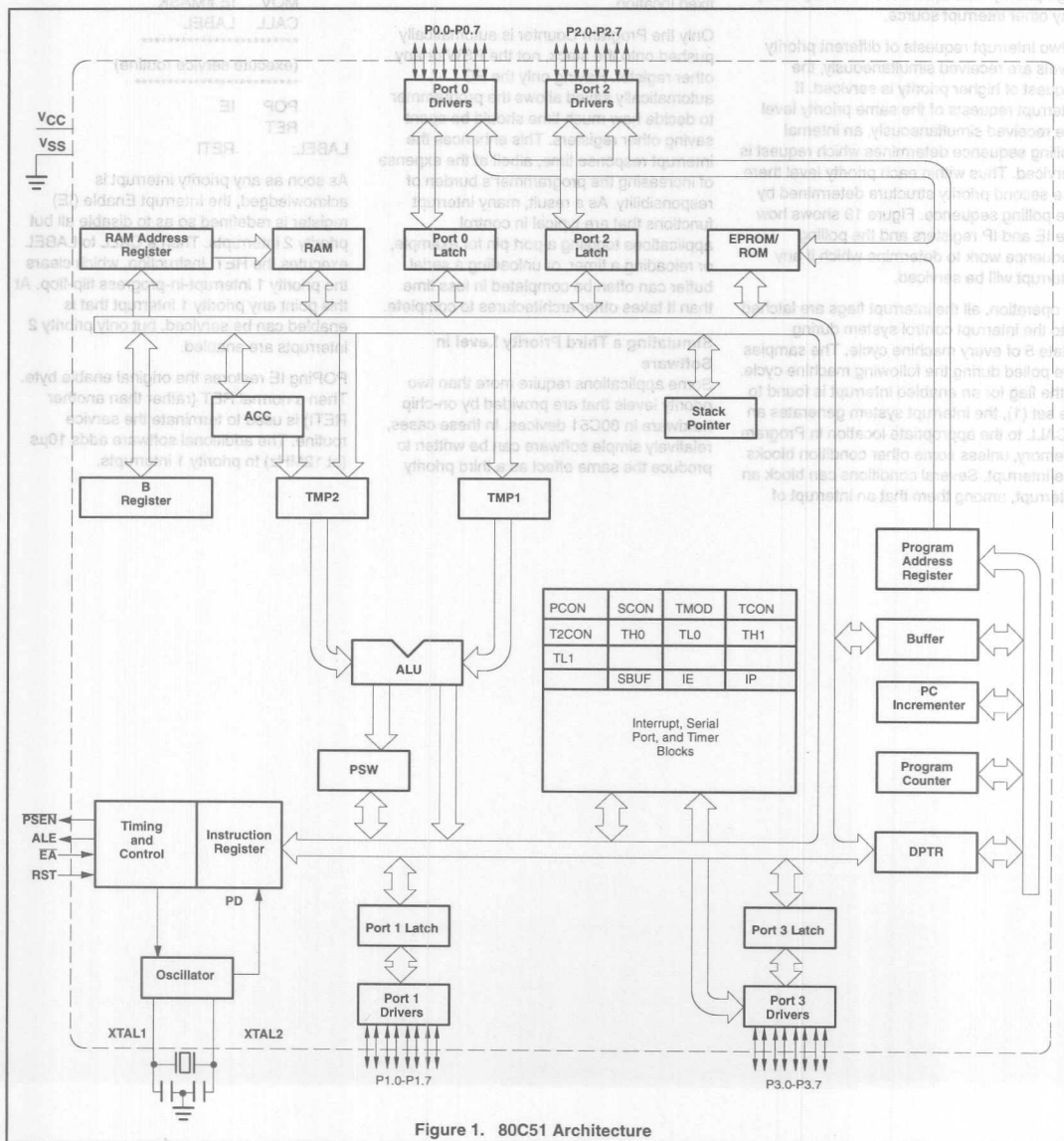


Figure 1. 80C51 Architecture

## 80C51 Family

## 80C51 family hardware description

## Special Function Registers

A Map of the on-chip memory area called the Special Function Register (SFR) space is shown in Figure 2.

Note that in the SFRs not all of the addresses are occupied. Unoccupied addresses are not implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have no effect.

User software should not write 1s to these unimplemented locations, since they may be used in other 80C51 Family derivative products to invoke new features. The functions of the SFRs are described in the text that follows.

## Accumulator

ACC is the Accumulator register. The mnemonics for Accumulator-Specific instructions, however, refer to the Accumulator simply as A.

## B Register

The B register is used during multiply and divide operations. For other instructions it can be treated as another scratch pad register.

## Program Status Word

The PSW register contains program status information as detailed in Figure 3.

## Stack Pointer

The Stack Pointer register is 8 bits wide. It is

incremented before data is stored during PUSH and CALL executions. While the stack may reside anywhere in on-chip RAM, the Stack Pointer is initialized to 07H after a reset. This causes the stack to begin at locations 08H.

## Data Pointer

The Data Pointer (DPTR) consists of a high byte (DPH) and a low byte (DPL). Its intended function is to hold a 16-bit address. It may be manipulated as a 16-bit register or as two independent 8-bit registers.

## Ports 0 to 3

P0, P1, P2, and P3 are the SFR latches of Ports 0, 1, 2, and 3, respectively. Writing a one to a bit of a port SFR (P0, P1, P2, or P3) causes the corresponding port output pin to switch high. Writing a zero causes the port output pin to switch low. When used as an input, the external state of a port pin will be held in the port SFR (i.e., if the external state of a pin is low, the corresponding port SFR bit will contain a 0; if it is high, the bit will contain a 1).

## Serial Data Buffer

The Serial Buffer is actually two separate registers, a transmit buffer and a receive buffer. When data is moved to SBUF, it goes to the transmit buffer and is held for serial transmission. (Moving a byte to SBUF is what

initiates the transmission.) When data is moved from SBUF, it comes from the receive buffer.

## Timer Registers Basic to 80C51

Register pairs (TH0, TL0), and (TH1, TL1) are the 16-bit Counting registers for Timer/Counters 0 and 1, respectively.

## Control Register for the 80C51

Special Function Registers IP, IE, TMOD, TCON, SCON, and PCON contain control and status bits for the interrupt system, the Timer/Counters, and the serial port. They are described in later sections.

## Port Structures and Operation

All four ports in the 80C51 are bidirectional. Each consists of a latch (Special Function Registers P0 through P3), an output driver, and an input buffer.

The output drivers of Ports 0 and 2, and the input buffers of Port 0, are used in accesses to external memory. In this application, Port 0 outputs the low byte of the external memory address, time-multiplexed with the byte being written or read. Port 2 outputs the high byte of the external memory address when the address is 16 bits wide. Otherwise, the Port 2 pins continue to emit the P2 SFR content.

8 Bytes									
F8								FF	
F0	B							F7	
E8								E7	
E0	ACC							E6	
D8								D7	
D0	PSW							D6	
C8								C7	
C0								C6	
B8	IP							B7	
B0	P3							B6	
A8	IE							A7	
A0	P2							A6	
98	SCON	SBUF						9F	
90	P1							97	
88	TCON	TMOD	TL0	TL1	TH0	TH1		8F	
80	P0	SP	DPL	DPH				87	PCON

Bit Addressable

Figure 2. 80C51 SFR Memory Map



## 80C51 Family

## 80C51 family hardware description

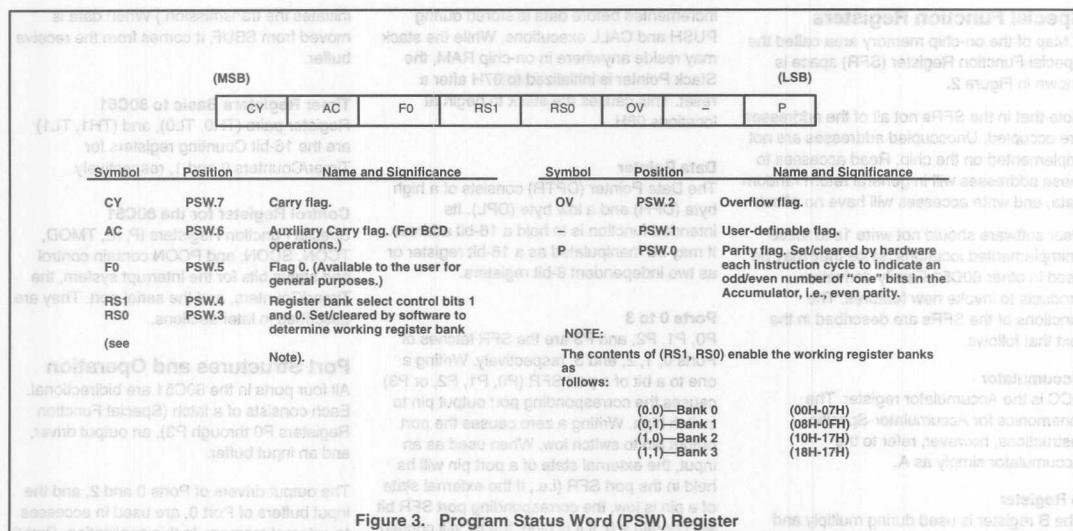


Figure 3. Program Status Word (PSW) Register

All the Port 3 pins are multifunctional. They are not only port pins, but also serve the functions of various special features as listed below:

**Port**

Pin	Alternate Function
P3.0	RxD (serial input port)
P3.1	TxD (serial output port)
P3.2	INT0 (external interrupt)
P3.3	INT1 (external interrupt)
P3.4	T0 (Timer/Counter 0 external input)
P3.5	T1 (Timer/Counter 1 external input)
P3.6	WR (external Data Memory write strobe)
P3.7	RD (external Data Memory read strobe)

The alternate functions can only be activated if the corresponding bit latch in the port SFR contains a 1. Otherwise the port pin remains at 0.

**I/O Configurations**

Figure 4 shows a functional diagram of a typical bit latch and I/O buffer in each of the four ports. The bit latch (one bit in the port's SFR) is represented as a Type D flip-flop, which will clock in a value from the internal bus in response to a "write to latch" signal from the CPU. The level of the port pin itself is placed on the internal bus in response to a "read pin" signal from the CPU. Some instructions that read a port activate the "read latch" signal, and others activate the "read pin" signal.

As shown in Figure 4, the output drivers of Port 0 and 2 are switchable to an internal

ADDR and ADDR/DATA bus by an internal CONTROL signal for use in external memory accesses. During external memory accesses, the P2 SFR remains unchanged, but the P0 SFR gets 1s written to it.

Also shown in Figure 4 is that if a P3 bit latch contains a 1, then the output level is controlled by the signal labeled "alternate output function." The actual P3.X pin level is always available to the pin's alternate input function, if any.

Ports 1, 2, and 3 have internal pullups, and Port 0 has open drain outputs. Each I/O line can be independently used as an input or an output. (Port 0 and 2 may not be used as general purpose I/O when being used as the ADDR/DATA BUS for external memory during normal operation.) To be used as an input, the port bit latch must contain a 1, which turns off the output driver FET. Then, for Ports 1, 2, and 3, the pin is pulled high by a weak internal pullup, and can be pulled low by an external source.

Port 0 differs in that its internal pullups are not active during normal port operation. The pullup FET in the P0 output driver (see Figure 4) is used only when the port is emitting 1s during external memory accesses. Otherwise the pullup FET is off. Consequently P0 lines that are being used as output port lines are open drain. Writing a 1 to the bit latch leaves both output FETs off, so the pin floats. In that condition it can be used as a high-impedance input.

Because Ports 1, 2, and 3 have fixed internal pullups, they are sometimes called

"quasi-bidirectional" ports. When configured as inputs they pull high and will source current (I<sub>IL</sub> in the data sheets) when externally pulled low. Port 0, on the other hand, is considered "true" bidirectional, because when configured as an input it floats.

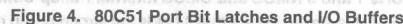
All the port latches in the 80C51 have 1s written to them by the reset function. If a 0 is subsequently written to a port latch, it can be reconfigured as an input by writing a 1 to it.

**Writing to a Port**

In the execution of an instruction that changes the value in a port latch, the new value arrives at the latch during S6P2 of the final cycle of the instruction. However, port latches are in fact sampled by their output buffers only during Phase 1 of an clock period. (During Phase 2 the output buffer holds the value it saw during the previous Phase 1). Consequently, the new value in the port latch won't actually appear at the output pin until the next Phase 1, which will be at S1P1 of the next machine cycle.

If the change requires a 0-to-1 transition in Port 1, 2, or 3, an additional pullup is turned on during S1P1 and S1P2 of the cycle in which the transition occurs. This is done to increase the transition speed. The extra pullup can source about 100 times the current that the normal pullup can. It should be noted that the internal pullups are field-effect transistors, not linear resistors. The pullup arrangements are shown in Figure 5.

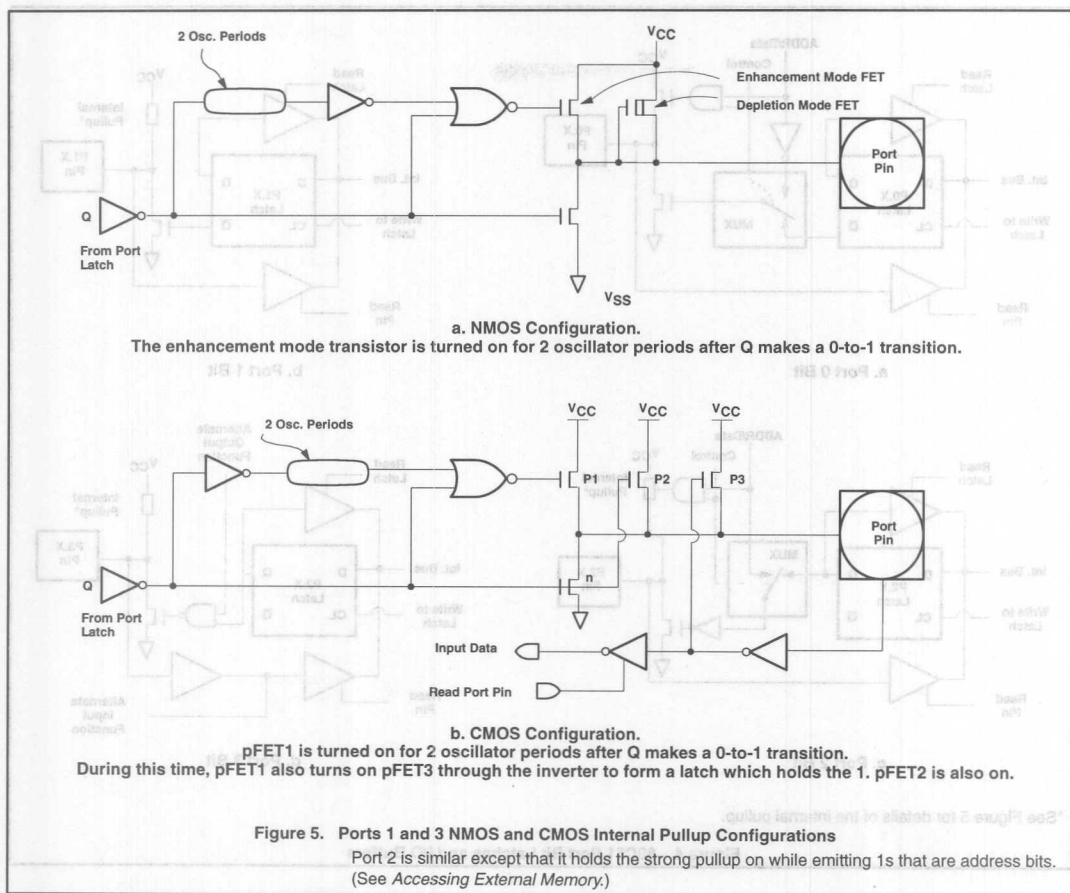
## 80C51 family hardware description



Port 0 output buffers can each drive 8 LS TTL inputs. They do, however, require external pullups to drive NMOS inputs, except when being used as the ADDRESS/DATA bus for external memory.

## 80C51 Family

## 80C51 family hardware description



## Read-Modify-Write Feature

Some instructions that read a port read the latch and others read the pin. Which ones do which? The instructions that read the latch rather than the pin are the ones that read a value, possibly change it, and then rewrite it to the latch. These are called "read-modify-write" instructions. The instructions listed below are read-modify-write instructions. When the destination operand is a port, or a port bit, these instructions read the latch rather than the pin:

ANL (logical AND, e.g., ANL P1,A)  
ORL (logical OR, e.g., ORL P2,A)  
XRL (logical EX-OR, e.g., XRL P3,A)  
JBC (jump if bit = 1 and clear bit, e.g., JBC P1.1,LABEL)  
CPL (complement bit, e.g.,

CPL P3.0)  
INC (increment, e.g., INC P2)  
DEC (decrement, e.g., DEC P2)  
DJNZ (decrement and jump if not zero, e.g., DJNZ P3,LABEL)  
MOV PX,Y,C (move carry bit to bit Y of Port X)  
CLR PX,Y (clear bit Y of Port X)  
SET PX,Y (set bit Y of Port X)

It is not obvious that the last three instructions in this list are read-modify-write instructions, but they are. They read the port byte, all 8 bits, modify the addressed bit, then write the new byte back to the latch.

The reason that read-modify-write instructions are directed to the latch rather than the pin is to avoid a possible misinterpretation of the voltage level at the pin. For example, a port bit might be used to

drive the base of a transistor. When a 1 is written to the bit, the transistor is turned on. If the CPU then reads the same port bit at the pin rather than the latch, it will read the base voltage of the transistor and interpret it as a 0. Reading the latch rather than the pin will return the correct value of 1.

## 80C51 Family

## 80C51 family hardware description

**Accessing External Memory**

Accesses to external memory are of two types: accesses to external Program Memory and accesses to external Data Memory. Accesses to external Program Memory use signal PSEN (program store enable) as the read strobe. Accesses to external Data Memory use RD or WR (alternate functions of P3.7 and P3.6) to strobe the memory. Fetches from external Program Memory always use a 16-bit address. Accesses to external Data Memory can use either a 16-bit address (MOVX @DPTR) or an 8-bit address (MOVX @Ri).

Whenever a 16-bit address is used, the high byte of the address comes out on Port 2, where it is held for the duration of the read or write cycle. Note that the Port 2 drivers use the strong pullups during the entire time that they are emitting address bits that are 1s. This is during the execution of a MOVX @DPTR instruction. During this time the Port 2 latch (the Special Function Register) does not have to contain 1s, and the contents of the Port 2 SFR are not modified. If the external memory cycle is not immediately followed by another external memory cycle, the undisturbed contents of the Port 2 SFR will reappear in the next cycle.

If an 8-bit address is being used (MOVX @Ri), the contents of the Port 2 SFR remain at the Port 2 pins throughout the external memory cycle. This will facilitate paging.

In any case, the low byte of the address is time-multiplexed with the data byte on Port 0. The ADDR/DATA signals drive both FETs in the Port 0 output buffers. Thus, in this application the Port 0 pins are not open-drain outputs, and do not require external pullups. ALE (Address Latch Enable) should be used to capture the address byte into an external latch. The address byte is valid at the negative transition of ALE. Then, in a write cycle, the data byte to be written appears on Port 0 just before WR is activated, and remains there until after WR is deactivated. In a read cycle, the incoming byte is accepted at Port 0 just before the read strobe is deactivated.

During any access to external memory, the CPU writes 0FFH to the Port 0 latch (the Special Function Register), thus obliterating whatever information the Port 0 SFR may have been holding.

External Program Memory is accessed under two conditions: Whenever signal EA is active; or whenever the program counter (PC) contains a number that is larger than 0FFFH (in the 80C51).

This requires that the ROMless versions have EA wired low to enable the lower 4k program bytes to be fetched from external memory.

When the CPU is executing out of external Program Memory, all 8 bits of Port 2 are dedicated to an output function and may not be used for general purpose I/O. During external program fetches they output the high byte of the PC. During this time the Port 2 drivers use the strong pullups to emit PC bits that are 1s.

**Timer/Counters**

The 80C51 has two 16-bit Timer/Counter registers: Timer 0 and Timer 1. Both can be configured to operate either as timers or event counters (see Figure 6).

In the "Timer" function, the register is incremented every machine cycle. Thus, one can think of it as counting machine cycles. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In the "Counter" function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T0 or T1. In this function, the external input is sampled during S5P2 of every machine cycle.

When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since it takes 2 machine cycles (24 oscillator periods) to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it should be held for at least one full cycle. In addition to the "Timer" or "Counter" selection, Timer 0 and Timer 1 have four operating modes from which to select.

**Timer 0 and Timer 1**

The "Timer" or "Counter" function is selected by control bits C/T in the Special Function Register TMOD. These two Timer/Counters have four operating modes, which are selected by bit-pairs (M1, M0) in TMOD. Modes 0, 1, and 2 are the same for both Timers/Counters. Mode 3 is different. The four operating modes are described in the following text.

**Mode 0**

Putting either Timer into Mode 0 makes it look like an 8048 Timer, which is an 8-bit Counter with a divide-by-32 prescaler.

Figure 7 shows the Mode 0 operation as it applies to Timer 1.

In this mode, the Timer register is configured as a 13-bit register. As the count rolls over from all 1s to all 0s, it sets the Timer interrupt flag TF1. The counted input is enabled to the Timer when TR1 = 1 and either GATE = 0 or INT1 = 1. (Setting GATE = 1 allows the Timer to be controlled by external input INT1, to facilitate pulse width measurements). TR1 is a control bit in the Special Function Register TCON (Figure 8). GATE is in TMOD.

The 13-bit register consists of all 8 bits of TH1 and the lower 5 bits of TL1. The upper 3 bits of TL1 are indeterminate and should be ignored. Setting the run flag (TR1) does not clear the registers.

Mode 0 operation is the same for the Timer 0 as for Timer 1. Substitute TR0, TF0, and INT0 for the corresponding Timer 1 signals in Figure 7. There are two different GATE bits, one for Timer 1 (TMOD.7) and one for Timer 0 (TMOD.3).

**Mode 1**

Mode 1 is the same as Mode 0, except that the Timer register is being run with all 16 bits.

**Mode 2**

Mode 2 configures the Timer register as an 8-bit Counter (TL1) with automatic reload, as shown in Figure 9. Overflow from TL1 not only sets TF1, but also reloads TL1 with the contents of TH1, which is preset by software. The reload leaves TH1 unchanged.

Mode 2 operation is the same for Timer/Counter 0.

**Mode 3**

Timer 1 in Mode 3 simply holds its count. The effect is the same as setting TR1 = 0.

Timer 0 in Mode 3 establishes TL0 and TH0 as two separate counters. The logic for Mode 3 on Timer 0 is shown in Figure 10. TL0 uses the Timer 0 control bits: C/T, GATE, TR0, INT0, and TF0. TH0 is locked into a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from Timer 1. Thus, TH0 now controls the "Timer 1" interrupt.

Mode 3 is provided for applications requiring an extra 8-bit timer on the counter. With Timer 0 in Mode 3, an 80C51 can look like it has three Timer/Counters. When Timer 0 is in Mode 3, Timer 1 can be turned on and off by switching it out of and into its own Mode 3, or can still be used by the serial port as a baud rate generator, or in fact, in any application not requiring an interrupt.

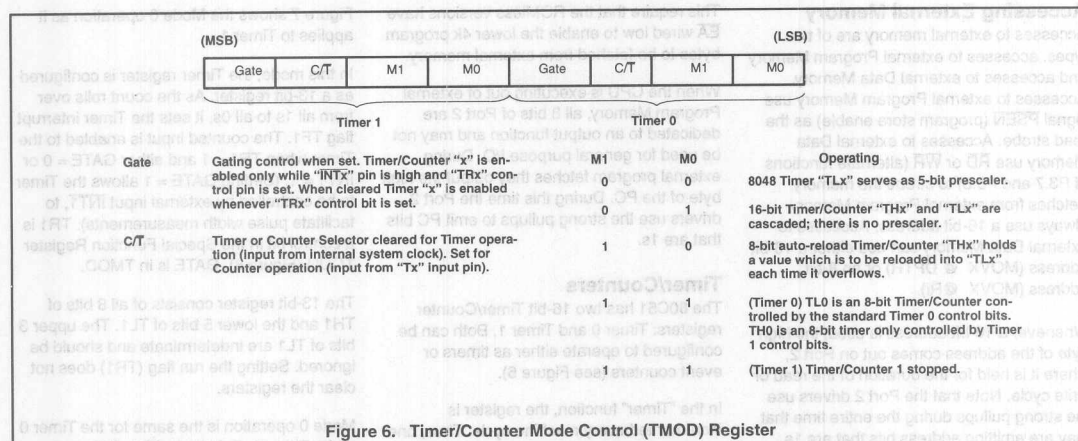


Figure 6. Timer/Counter Mode Control (TMOD) Register

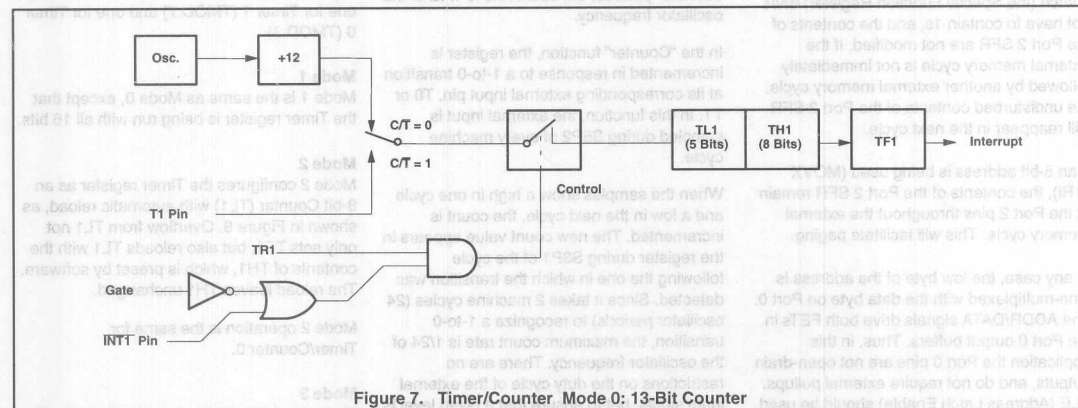


Figure 7. Timer/Counter Mode 0: 13-Bit Counter

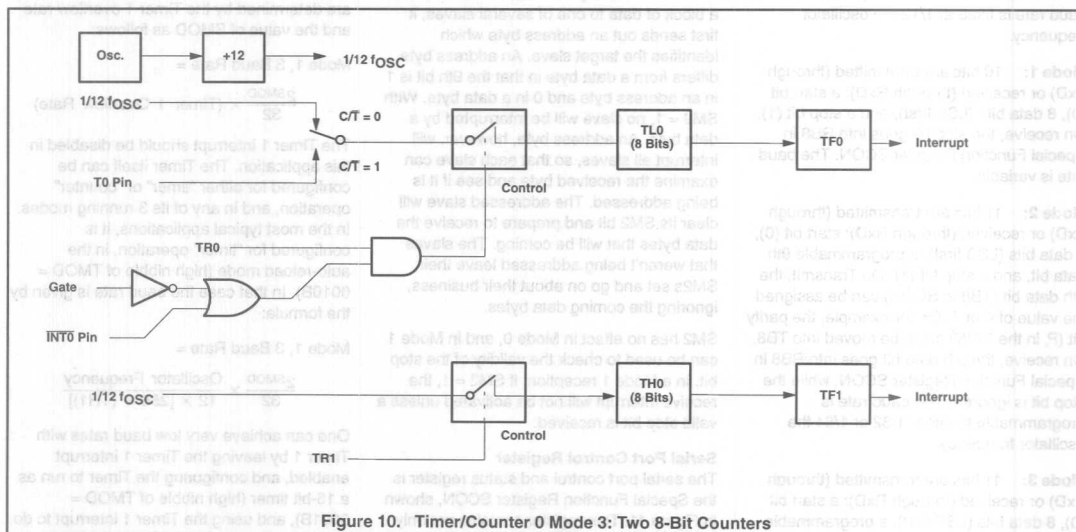
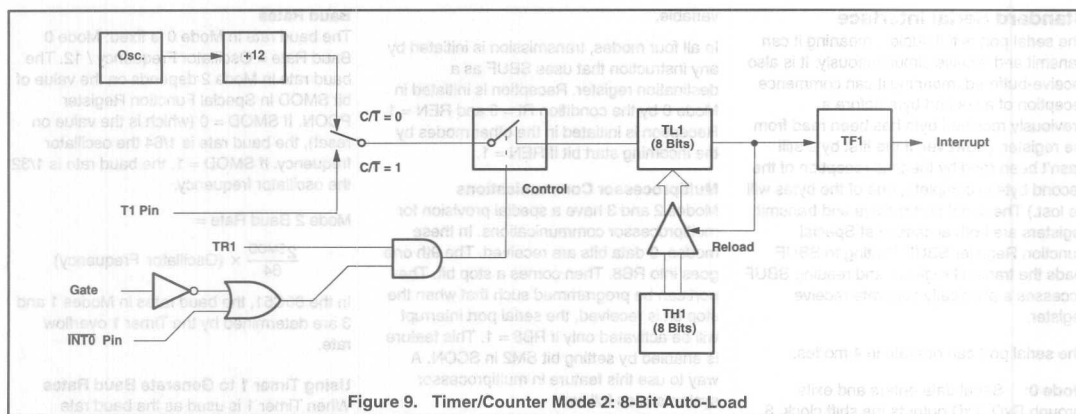
(MSB)				(LSB)			
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
Symbol	Position	Name and Significance		Symbol	Position	Name and Significance	
TF1	TCON.7	Timer 1 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine.		IE1	TCON.3	Interrupt 1 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.	
TR1	TCON.6	Timer 1 Run control bit. Set/cleared by software to turn Timer/Counter on/off.		IT1	TCON.2	Interrupt 1 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.	
TF0	TCON.5	Timer 0 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine.		IE0	TCON.1	Interrupt 0 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.	
TR0	TCON.4	Timer 0 Run control bit. Set/cleared by software to turn Timer/Counter on/off.		IT0	TCON.0	Interrupt 0 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.	

Figure 8. Timer/Counter Control (TCON) Register



## 80C51 Family

## 80C51 family hardware description





## 80C51 Family

## 80C51 family hardware description

**Standard Serial Interface**

The serial port is full duplex, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the register. (However, if the first byte still hasn't been read by the time reception of the second byte is complete, one of the bytes will be lost.) The serial port receive and transmit registers are both accessed at Special Function Register SBUF. Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register.

The serial port can operate in 4 modes:

**Mode 0:** Serial data enters and exits through Rx/D. Tx/D outputs the shift clock. 8 bits are transmitted/received (LSB first). The baud rate is fixed at 1/12 the oscillator frequency.

**Mode 1:** 10 bits are transmitted (through Tx/D) or received (through Rx/D): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in Special Function Register SCON. The baud rate is variable.

**Mode 2:** 11 bits are transmitted (through Tx/D) or received (through Rx/D): start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On Transmit, the 9th data bit (TB8 in SCON) can be assigned the value of 0 or 1. Or, for example, the parity bit (P, in the PSW) could be moved into TB8. On receive, the 9th data bit goes into RB8 in Special Function Register SCON, while the stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 the oscillator frequency.

**Mode 3:** 11 bits are transmitted (through Tx/D) or received (through Rx/D): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). In fact, Mode 3 is the same as Mode 2 in all respects except baud rate. The baud rate in Mode 3 is

variable.

In all four modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit if REN = 1.

**Multiprocessor Communications**

Modes 2 and 3 have a special provision for multiprocessor communications. In these modes, 9 data bits are received. The 9th one goes into RB8. Then comes a stop bit. The port can be programmed such that when the stop bit is received, the serial port interrupt will be activated only if RB8 = 1. This feature is enabled by setting bit SM2 in SCON. A way to use this feature in multiprocessor systems is as follows:

When the master processor wants to transmit a block of data to one of several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte. With SM2 = 1, no slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave will clear its SM2 bit and prepare to receive the data bytes that will be coming. The slaves that weren't being addressed leave their SM2s set and go on about their business, ignoring the coming data bytes.

SM2 has no effect in Mode 0, and in Mode 1 can be used to check the validity of the stop bit. In a Mode 1 reception, if SM2 = 1, the receive interrupt will not be activated unless a valid stop bit is received.

**Serial Port Control Register**

The serial port control and status register is the Special Function Register SCON, shown in Figure 11. This register contains not only the mode selection bits, but also the 9th data bit for transmit and receive (TB8 and RB8), and the serial port interrupt bits (TI and RI).

**Baud Rates**

The baud rate in Mode 0 is fixed: Mode 0 Baud Rate = Oscillator Frequency / 12. The baud rate in Mode 2 depends on the value of bit SMOD in Special Function Register PCON. If SMOD = 0 (which is the value on reset), the baud rate is 1/64 the oscillator frequency. If SMOD = 1, the baud rate is 1/32 the oscillator frequency.

Mode 2 Baud Rate =

$$\frac{2^{\text{SMOD}}}{64} \times (\text{Oscillator Frequency})$$

In the 80C51, the baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate.

**Using Timer 1 to Generate Baud Rates**

When Timer 1 is used as the baud rate generator, the baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate and the value of SMOD as follows:

Mode 1, 3 Baud Rate =

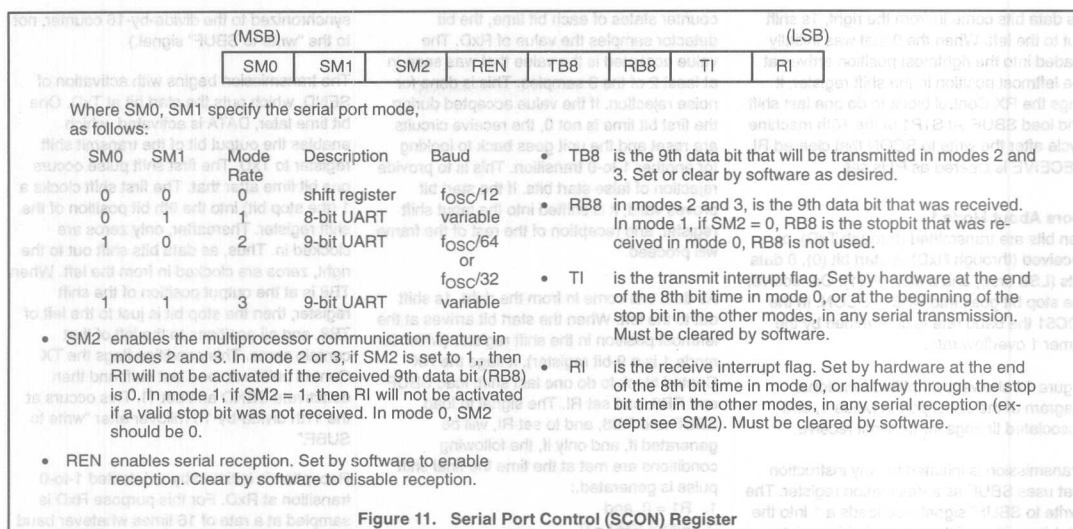
$$\frac{2^{\text{SMOD}}}{32} \times (\text{Timer 1 Overflow Rate})$$

The Timer 1 interrupt should be disabled in this application. The Timer itself can be configured for either "timer" or "counter" operation, and in any of its 3 running modes. In the most typical applications, it is configured for "timer" operation, in the auto-reload mode (high nibble of TMOD = 0010B). In that case the baud rate is given by the formula:

Mode 1, 3 Baud Rate =

$$\frac{2^{\text{SMOD}}}{32} \times \frac{\text{Oscillator Frequency}}{12 \times [256 - (\text{TH1})]}$$

One can achieve very low baud rates with Timer 1 by leaving the Timer 1 interrupt enabled, and configuring the Timer to run as a 16-bit timer (high nibble of TMOD = 0001B), and using the Timer 1 interrupt to do a 16-bit software reload. Figure 12 lists various commonly used baud rates and how they can be obtained from Timer 1.



Baud Rate	f <sub>osc</sub>	SMOD	C/T	Timer 1	
				Mode	Reload Value
Mode 0 Max: 1.67MHz	20MHz	X	X	X	X
Mode 2 Max: 625k	20MHz	X	X	X	X
Mode 1, 3 Max: 104.2k	20MHz	1	0	2	FFH
19.2k	11.059MHz	1	0	2	FDH
9.6k	11.059MHz	0	0	2	FDH
4.8k	11.059MHz	0	0	2	FAH
2.4k	11.059MHz	0	0	2	F4H
1.2k	11.059MHz	0	0	2	E8H
137.5	11.986M	0	0	2	1DH
110	6MHz	0	0	2	72H
110	12MHz	0	0	1	FEEDH

**Figure 12. Timer 1 Generated Commonly Used Baud Rates**

### More About Mode 0

Serial data enters and exits through Rx/D. Tx/D outputs the shift clock. 8 bits are transmitted/received: 8 data bits (LSB first). The baud rate is fixed at 1/12 the oscillator frequency.

Figure 13 shows a simplified functional diagram of the serial port in Mode 0, and associated timing.

Transmission is initiated by any instruction that uses SBUF as a destination register. The "write to SBUF" signal at S6P2 also loads a 1 into the 9th position of the transmit shift register and tells the TX Control block to commence a transmission. The internal timing is such that one full machine cycle will elapse between "write to SBUF" and

activation of SEND.

SEND enables the output of the shift register to the alternate output function line of P3.0 and also enable SHIFT CLOCK to the alternate output function line of P3.1. SHIFT CLOCK is low during S3, S4, and S5 of every machine cycle, and high during S6, S1, and S2. At S6P2 of every machine cycle in which SEND is active, the contents of the transmit shift are shifted to the right one position.

As data bits shift out to the right, zeros come in from the left. When the MSB of the data byte is at the output position of the shift register, then the 1 that was initially loaded into the 9th position, is just to the left of the MSB, and all positions to the left of that contain zeros. This condition flags the TX Control block to do one last shift and then

deactivate SEND and set TI. Both of these actions occur at S1P1 of the 10th machine cycle after "write to SBUF."

Reception is initiated by the condition REN = 1 and RI = 0. At S6P2 of the next machine cycle, the RX Control unit writes the bits 11111110 to the receive shift register, and in the next clock phase activates RECEIVE.

RECEIVE enable SHIFT CLOCK to the alternate output function line of P3.1. SHIFT CLOCK makes transitions at S3P1 and S6P1 of every machine cycle. At S6P2 of every machine cycle in which RECEIVE is active, the contents of the receive shift register are shifted to the left one position. The value that comes in from the right is the value that was sampled at the P3.0 pin at S5P2 of the same machine cycle.

## 80C51 Family

## 80C51 family hardware description

As data bits come in from the right, 1s shift out to the left. When the 0 that was initially loaded into the rightmost position arrives at the leftmost position in the shift register, it flags the RX Control block to do one last shift and load SBUF. At S1P1 of the 10th machine cycle after the write to SCON that cleared RI, RECEIVE is cleared as RI is set.

**More About Mode 1**

Ten bits are transmitted (through TxD), or received (through RxD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in SCON. In the 80C51 the baud rate is determined by the Timer 1 overflow rate.

Figure 14 shows a simplified functional diagram of the serial port in Mode 1, and associated timings for transmit/receive.

Transmission is initiated by any instruction that uses SBUF as a destination register. The "write to SBUF" signal also loads a 1 into the 9th bit position of the transmit shift register and flags the TX Control unit that a transmission is requested. Transmission actually commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter. (Thus, the bit times are synchronized to the divide-by-16 counter, not to the "write to SBUF" signal.)

The transmission begins with activation of SEND which puts the start bit at TxD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TxD. The first shift pulse occurs one bit time after that.

As data bits shift out to the right, zeros are clocked in from the left. When the MSB of the data byte is at the output position of the shift register, then the 1 that was initially loaded into the 9th position is just to the left of the MSB, and all positions to the left of that contain zeros. This condition flags the TX Control unit to do one last shift and then deactivate SEND and set TI. This occurs at the 10th divide-by-16 rollover after "write to SBUF."

Reception is initiated by a detected 1-to-0 transition at RxD. For this purpose RxD is sampled at a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FFH is written into the input shift register. Resetting the divide-by-16 counter aligns its rollovers with the boundaries of the incoming bit times.

The 16 states of the counter divide each bit time into 16ths. At the 7th, 8th, and 9th

counter states of each bit time, the bit detector samples the value of RxD. The value accepted is the value that was seen in at least 2 of the 3 samples. This is done for noise rejection. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back to looking for another 1-to-0 transition. This is to provide rejection of false start bits. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

As data bits come in from the right, 1s shift out to the left. When the start bit arrives at the leftmost position in the shift register (which in mode 1 is a 9-bit register), it flags the RX Control block to do one last shift, load SBUF and RB8, and set RI. The signal to load SBUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated:

1. R1 = 0, and
2. Either SM2 = 0, or the received stop bit = 1.

If either of these two conditions is not met, the received frame is irretrievably lost. If both conditions are met, the stop bit goes into RB8, the 8 data bits go into SBUF, and RI is activated. At this time, whether the above conditions are met or not, the unit goes back to looking for a 1-to-0 transition in RxD.

**More About Modes 2 and 3**

Eleven bits are transmitted (through TxD), or received (through RxD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On transmit, the 9th data bit (TB8) can be assigned the value of 0 or 1. On receive, the 9th data bit goes into RB8 in SCON. The baud rate is programmable to either 1/32 or 1/64 the oscillator frequency in Mode 2. Mode 3 may have a variable baud rate generated from Timer 1.

Figures 15 and 16 show a functional diagram of the serial port in Modes 2 and 3. The receive portion is exactly the same as in Mode 1. The transmit portion differs from Mode 1 only in the 9th bit of the transmit shift register.

Transmission is initiated by any instruction that uses SBUF as a destination register. The "write to SBUF" signal also loads TB8 into the 9th bit position of the transmit shift register and flags the TX Control unit that a transmission is requested. Transmission commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter. (Thus, the bit times are

synchronized to the divide-by-16 counter, not to the "write to SBUF" signal.)

The transmission begins with activation of SEND, which puts the start bit at TxD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TxD. The first shift pulse occurs one bit time after that. The first shift clocks a 1 (the stop bit) into the 9th bit position of the shift register. Thereafter, only zeros are clocked in. Thus, as data bits shift out to the right, zeros are clocked in from the left. When TB8 is at the output position of the shift register, then the stop bit is just to the left of TB8, and all positions to the left of that contain zeros. This condition flags the TX Control unit to do one last shift and then deactivate SEND and set TI. This occurs at the 11th divide-by-16 rollover after "write to SBUF."

Reception is initiated by a detected 1-to-0 transition at RxD. For this purpose RxD is sampled at a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FFH is written to the input shift register.

At the 7th, 8th, and 9th counter states of each bit time, the bit detector samples the value of R-D. The value accepted is the value that was seen in at least 2 of the 3 samples. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back to looking for another 1-to-0 transition. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

As data bits come in from the right, 1s shift out to the left. When the start bit arrives at the leftmost position in the shift register (which in Modes 2 and 3 is a 9-bit register), it flags the RX Control block to do one last shift, load SBUF and RB8, and set RI.

The signal to load SBUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated.

1. R1 = 0, and
2. Either SM2 = 0, or the received 9th data bit = 1.

If either of these conditions is not met, the received frame is irretrievably lost, and RI is not set. If both conditions are met, the received 9th data bit goes into RB8, and the first 8 data bits go into SBUF. One bit time later, whether the above conditions were met or not, the unit goes back to looking for a 1-to-0 transition at the RxD input.

## 80C51 Family

## 80C51 family hardware description

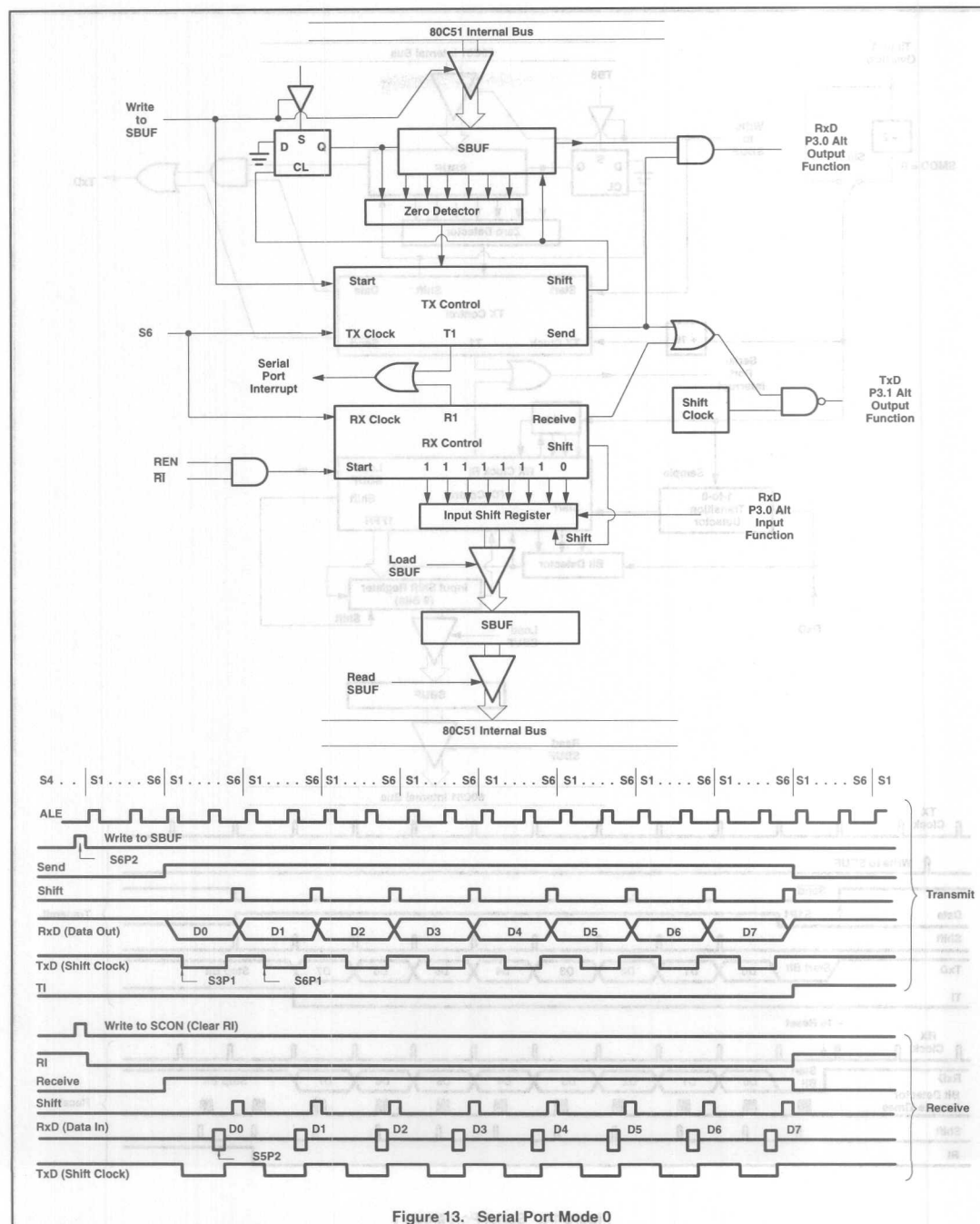


Figure 13. Serial Port Mode 0

## 80C51 Family

## 80C51 family hardware description

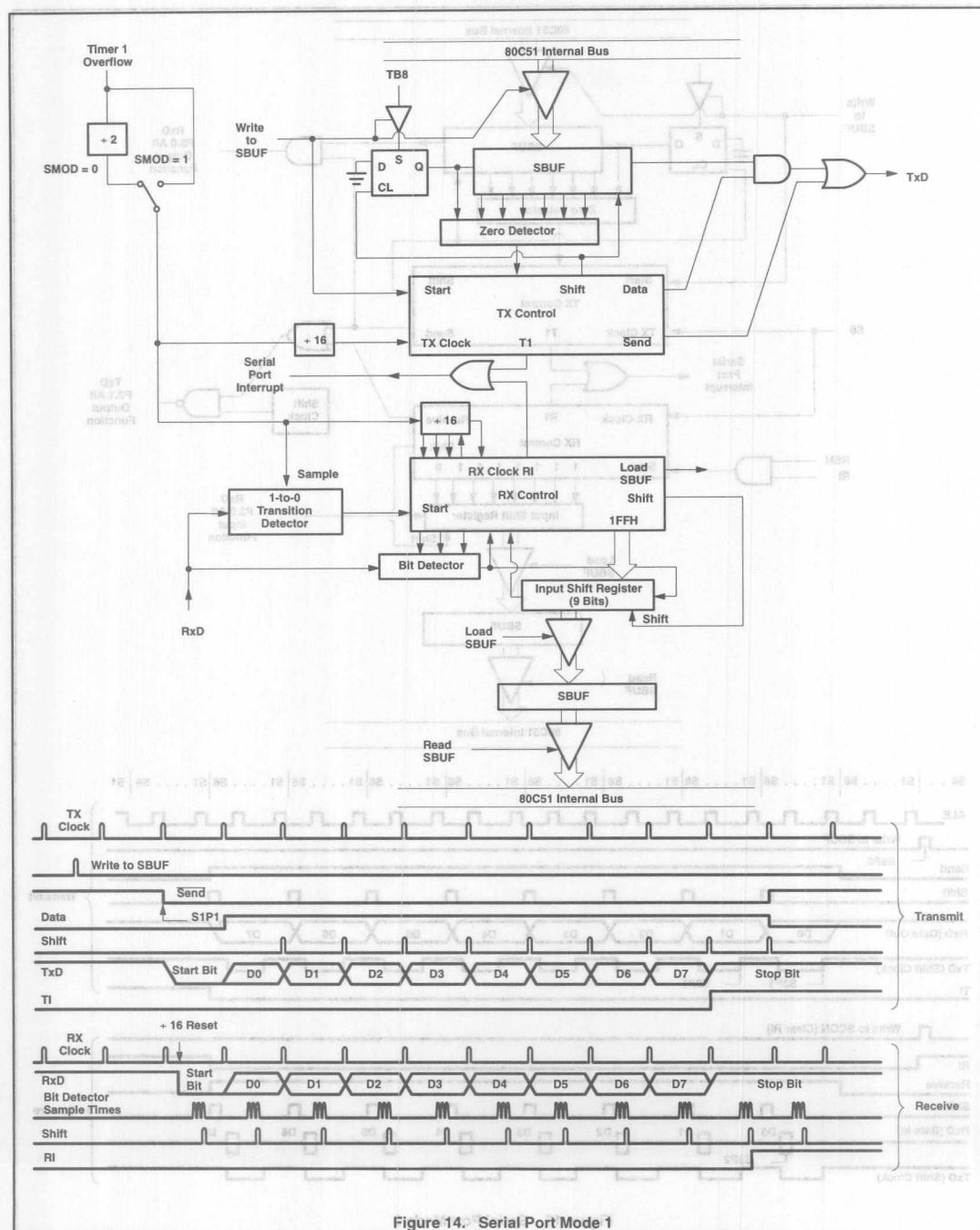


Figure 14. Serial Port Mode 1



# 80C51 Family

## 80C51 family hardware description

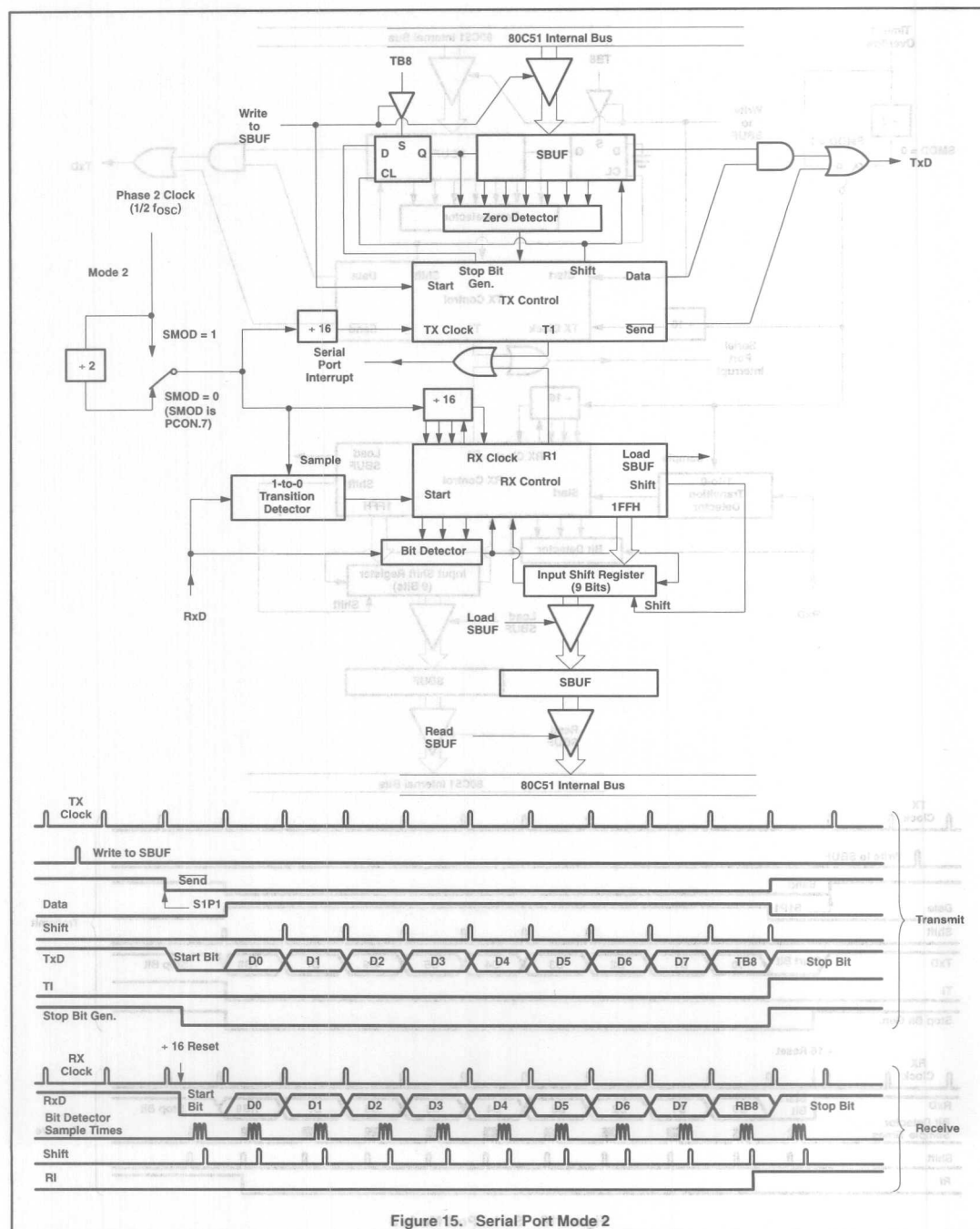


Figure 15. Serial Port Mode 2



# 80C51 Family

## 80C51 family hardware description

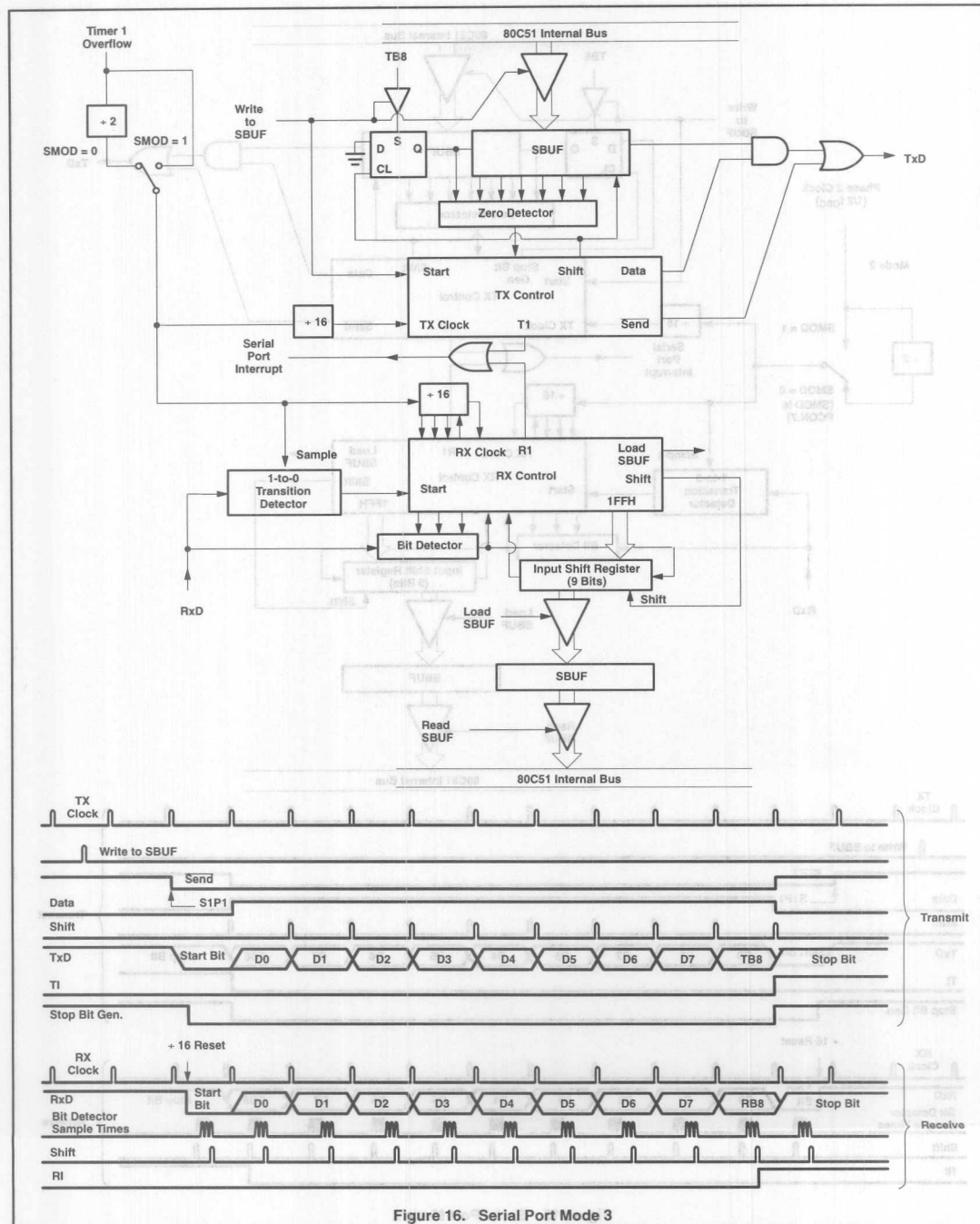


Figure 16. Serial Port Mode 3

## 80C51 Family

## 80C51 family hardware description

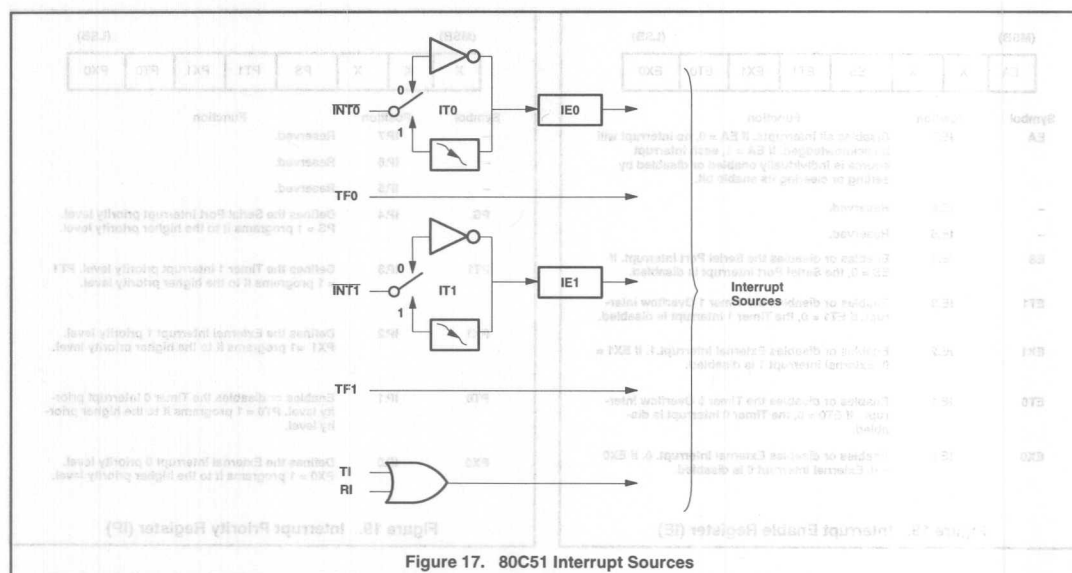


Figure 17. 80C51 Interrupt Sources

### Interrupts

The 80C51 provides 5 interrupt sources. These are shown in Figure 17. The External Interrupts INT0 and INT1 can each be either level-activated or transition-activated, depending on bits IT0 and IT1 in Register TCON. The flags that actually generate these interrupts are bits IE0 and IE1 in TCON. When an external interrupt is generated, the flag that generated it is cleared by the hardware when the service routine is vectored to only if the interrupt was transition-activated. If the interrupt was level-activated, then the external requesting source is what controls the request flag, rather than the on-chip hardware.

The Timer 0 and Timer 1 Interrupts are generated by TF0 and TF1, which are set by a rollover in their respective Timer/Counter registers (except see Timer 0 in Mode 3). When a timer interrupt is generated, the flag that generated it is cleared by the on-chip hardware when the service routine is vectored to.

The Serial Port Interrupt is generated by the logical OR of RI and TI. Neither of these flags

is cleared by hardware when the service routine is vectored to. In fact, the service routine will normally have to determine whether it was RI or TI that generated the interrupt, and the bit will have to be cleared in software.

All of the bits that generate interrupts can be set or cleared by software, with the same result as though it had been set or cleared by hardware. That is, interrupts can be generated or pending interrupts can be canceled in software.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE (Figure 18). IE also contains a global disable bit, EA, which disables all interrupts at once.

### Priority Level Structure

Each interrupt source can also be individually programmed to one of two priority levels by setting or clearing a bit in Special Function Register IP (Figure 19). A low-priority interrupt can itself be interrupted by a high-priority interrupt, but not by another low-priority interrupt. A high-priority interrupt

can't be interrupted by any other interrupt source.

If two request of different priority levels are received simultaneously, the request of higher priority level is serviced. If requests of the same priority level are received simultaneously, an internal polling sequence determines which request is serviced. Thus within each priority level there is a second priority structure determined by the polling sequence as follows:

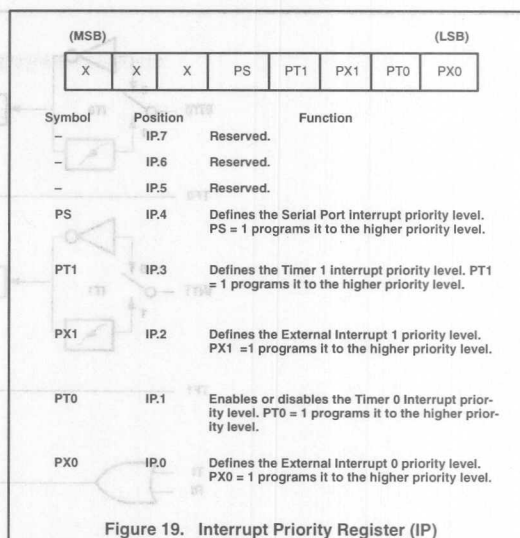
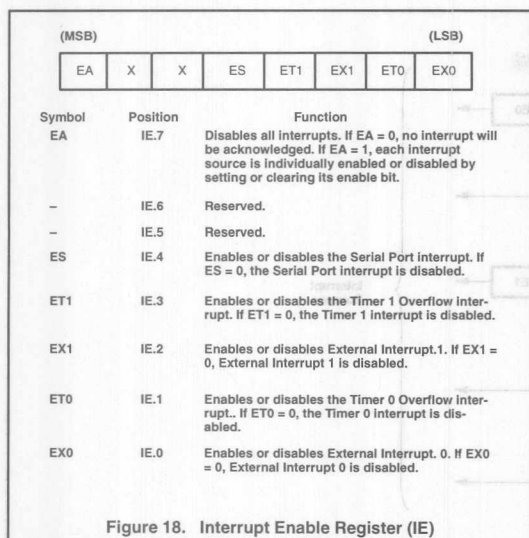
- | Source   | Priority Within Level |
|----------|-----------------------|
| 1. IE0   | (highest)             |
| 2. TF0   |                       |
| 3. IE1   |                       |
| 4. TF1   |                       |
| 5. RI+TI | (lowest)              |

Note that the "priority within level" structure is only used to resolve simultaneous requests of the same priority level.

The IP register contains a number of unimplemented bits. IP.7, IP.6, and IP.5 are reserved in the 80C51. User software should not write 1s to these positions, since they may be used in other 8051 Family products.

## 80C51 Family

## 80C51 family hardware description

**How Interrupts Are Handled**

The interrupt flags are sampled at S5P2 of every machine cycle. The samples are polled during the following machine cycle. If one of the flags was in a set condition at S5P2 of the preceding cycle, the polling cycle will find it and the interrupt system will generate an LCALL to the appropriate service routine, provided this hardware-generated LCALL is not blocked by any of the following conditions:

1. An interrupt of equal or higher priority level is already in progress.
2. The current (polling) cycle is not the final cycle in the execution of the instruction in progress.
3. The instruction in progress is RETI or any write to the IE or IP registers.

Any of these three conditions will block the generation of the LCALL to the interrupt service routine. Condition 2 ensures that the instruction in progress will be completed before vectoring to any service routine. Condition 3 ensures that if the instruction in progress is RETI or any access to IE or IP, then at least one more instruction will be executed before any interrupt is vectored to.

The polling cycle is repeated with each machine cycle, and the values polled are the values that were present at S5P2 of the previous machine cycle. Note that if an interrupt flag is active but not being responded to for one of the above conditions, if the flag is not still active when the blocking condition is removed, the denied interrupt will not be serviced. In other words, the fact that the interrupt flag was once active but not serviced is not remembered. Every polling cycle is new.

The polling cycle/LCALL sequence is illustrated in Figure 20.

Note that if an interrupt of higher priority level goes active prior to S5P2 of the machine cycle labeled C3 in Figure 20, then in accordance with the above rules it will be vectored to during C5 and C6, without any instruction of the lower priority routine having been executed.

Thus the processor acknowledges an interrupt request by executing a hardware-generated LCALL to the appropriate servicing routine. In some cases it also clears the flag that generated the interrupt, and in other cases it doesn't. It never clears the Serial Port flag. This has to be done in the user's software. It clears an

external interrupt flag (IE0 or IE1) only if it was transition-activated. The hardware-generated LCALL pushes the contents of the Program Counter on to the stack (but it does not save the PSW) and reloads the PC with an address that depends on the source of the interrupt being vectored to, as shown below:

**Source Vector Address**

IE0	0003H
TF0	000BH
IE1	0013H
TF1	001BH
RI+TI	0023H

Execution proceeds from that location until the RETI instruction is encountered. The RETI instruction informs the processor that this interrupt routine is no longer in progress, then pops the top two bytes from the stack and reloads the Program Counter. Execution of the interrupted program continues from where it left off.

Note that a simple RET instruction would also have returned execution to the interrupted program, but it would have left the interrupt control system thinking an interrupt was still in progress, making future interrupts impossible.

## 80C51 Family

## 80C51 family hardware description

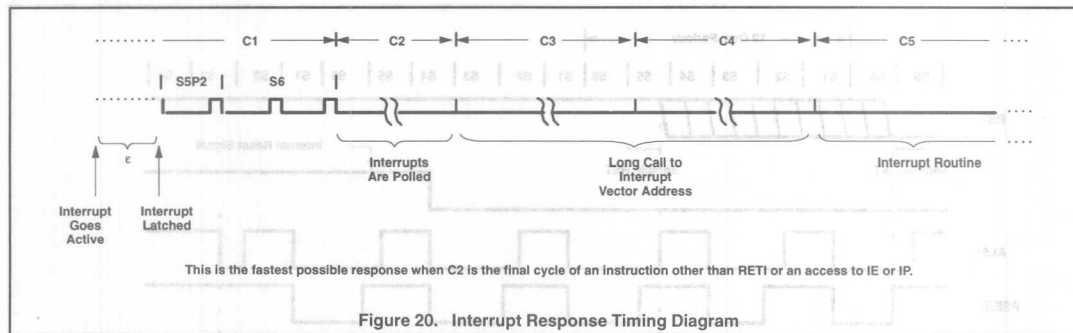


Figure 20. Interrupt Response Timing Diagram

## External Interrupts

The external sources can be programmed to be level-activated or transition-activated by setting or clearing bit IT1 or IT0 in Register TCON. If ITx = 0, external interrupt x is triggered by a detected low at the INTx pin. If ITx = 1, external interrupt x is edge triggered. In this mode if successive samples of the INTx pin show a high in one cycle and a low in the next cycle, interrupt request flag IEx in TCON is set. Flag bit IEx then requests the interrupt.

Since the external interrupt pins are sampled once each machine cycle, an input high or low should hold for at least 12 oscillator periods to ensure sampling. If the external interrupt is transition-activated, the external source has to hold the request pin high for at least one cycle, and then hold it low for at least one cycle. This is done to ensure that the transition is seen so that interrupt request flag IEx will be set. IEx will be automatically cleared by the CPU when the service routine is called.

If the external interrupt is level-activated, the external source has to hold the request active until the requested interrupt is actually generated. Then it has to deactivate the request before the interrupt service routine is completed, or else another interrupt will be generated.

## Response Time

The INT0 and INT1 levels are inverted and latched into IE0 and IE1 at S5P2 of every machine cycle. The values are not actually polled by the circuitry until the next machine cycle. If a request is active and conditions are right for it to be acknowledged, a hardware subroutine call to the requested service routine will be the next instruction to be executed. The call itself takes two cycles.

Thus, a minimum of three complete machine cycles elapse between activation of an external interrupt request and the beginning of execution of the first instruction of the service routine. Figure 20 shows interrupt response timings.

A longer response time would result if the request is blocked by one of the 3 previously listed conditions. If an interrupt of equal or higher priority level is already in progress, the additional wait time obviously depends on the nature of the other interrupt's service routine. If the instruction in progress is not in its final cycle, the additional wait time cannot be more than 3 cycles, since the longest instructions (MUL and DIV) are only 4 cycles long, and if the instruction in progress is RETI or an access to IE or IP, the additional wait time cannot be more than 5 cycles (a maximum of one more cycle to complete the instruction in progress, plus 4 cycles to complete the next instruction if the instruction is MUL or DIV).

Thus, in a single-interrupt system, the response time is always more than 3 cycles and less than 9 cycles.

## Single-Step Operation

The 80C51 interrupt structure allows single-step execution with very little software overhead. As previously noted, an interrupt request will not be responded to while an interrupt of equal priority level is still in progress, nor will it be responded to after RETI until at least one other instruction has been executed. Thus, once an interrupt routine has been entered, it cannot be re-entered until at least one instruction of the interrupted program is executed. One way to use this feature for single-step operation is to program one of the external interrupts (e.g., INT0) to be level-activated. The service

routine for the interrupt will terminate with the following code:

```
JNB P3.2, $ ;Wait Till INT0 Goes High
JB P3.2, $ ;Wait Till INT0 Goes Low
RETI ;Go Back and Execute One Instruction
```

Now if the INT0 pin, which is also the P3.2 pin, is held normally low, the CPU will go right into the External Interrupt 0 routine and stay there until INT0 is pulsed (from low to high to low). Then it will execute RETI, go back to the task program, execute one instruction, and immediately re-enter the External Interrupt 0 routine to await the next pulsing of P3.2. One step of the task program is executed each time P3.2 is pulsed.

## Reset

The reset input is the RST pin, which is the input to a Schmitt Trigger. A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. The CPU responds by generating an internal reset, with the timing shown in Figure 21.

The external reset signal is asynchronous to the internal clock. The RST pin is sampled during State 5 Phase 2 of every machine cycle. The port pins will maintain their current activities for 19 oscillator periods after a logic 1 has been sampled at the RST pin; that is, for 19 to 31 oscillator periods after the external reset signal has been applied to the RST pin.

The internal reset algorithm writes 0s to all the SFRs except the port latches, the Stack Pointer, and SBUF. The port latches are initialized to FFH, the Stack Pointer to 07H, and SBUF is indeterminate. Table 1 lists the SFR reset values. The internal RAM is not affected by reset. On power up the RAM content is indeterminate.

## 80C51 Family

## 80C51 family hardware description

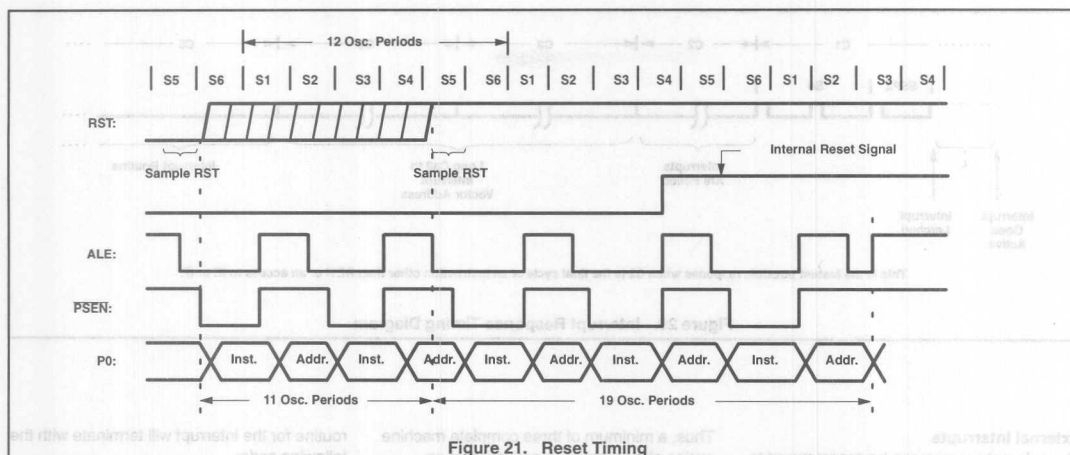


Figure 21. Reset Timing

Table 1. 80C51 SFR Reset Values

REGISTER	RESET VALUE
PC	000H
ACC	00H
B	00H
PSW	00H
SP	07H
DPTR	0000H
P0-P3	FFH
IP	XXX0000B
IE	0XX0000B
TMOD	00H
TCON	00H
TH0	00H
TL0	00H
TH1	00H
TL1	00H
SCON	00H
SBUF	Indeterminate
PCON (NMOS)	0XXXXXXB
PCON (CMOS)	0XXX0000B

## Power-on Reset

An automatic reset can be obtained when  $V_{CC}$  is turned on by connecting the RST pin to  $V_{CC}$  through a 10 $\mu$ F capacitor and to  $V_{SS}$  through an 8.2k resistor, providing the  $V_{CC}$  rise time does not exceed 1 millisecond and the oscillator start-up time does not exceed 10 milliseconds. This power-on reset circuit is shown in Figure 22. The CMOS devices do not require the 8.2k pulldown resistor, although its presence does no harm.

When power is turned on, the circuit holds the RST pin high for an amount of time that depends on the value of the capacitor and

the rate at which it charges. To ensure a good reset, the RST pin must be high long enough to allow the oscillator time to start-up (normally a few ms) plus two machine cycles.

*Note that the port pins will be in a random state until the oscillator has started and the internal reset algorithm has written 1s to them.*

With this circuit, reducing  $V_{CC}$  quickly to 0 causes the RST pin voltage to momentarily fall below 0V. However, this voltage is internally limited, and will not harm the device.

## Power-Saving Modes of Operation

For applications where power consumption is critical the CMOS version provides power reduced modes of operation as a standard feature. The power down mode in NMOS devices is no longer a standard feature.

## CMOS Power Reduction Mode

CMOS versions have two power reducing modes, Idle and Power Down. The input through which backup power is supplied during these operations is  $V_{CC}$ . Figure 23 shows the internal circuitry which implements these features. In the Idle modes (IDL = 1), the oscillator continues to run and the Interrupt, Serial Port, and Timer blocks continue to be clocked, but the clock signal is gated off to the CPU. In Power Down (PD = 1), the oscillator is frozen. The Idle and Power Down Modes are activated by setting bits in Special Function Register PCON. The address of this register is 87H. Figure 24 details its contents.

In the NMOS devices the PCON register only contains SMOD. The other four bits are implemented only in the CMOS devices. User software should never write 1s to unimplemented bits, since they may be used in other 80C51 Family products.

## Idle Mode

An instruction that sets PCON.0 causes that to be the last instruction executed before going into the Idle mode, the internal clock signal is gated off to the CPU but not to the Interrupt, Timer, and Serial Port functions. The CPU status is preserved in its entirety; the Stack Pointer, Program Counter, Program Status Word, Accumulator, and all other registers maintain their data during Idle. The port pins hold the logical states they had at the time Idle was activated. ALE and PSEN hold at logic high levels.

There are two ways to terminate the Idle. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, terminating the Idle mode. The interrupt will be serviced, and following RETI, the next instruction to be executed will be the one following the instruction that put the device into Idle.

The flag bits GF0 and GF1 can be used to give an indication if an interrupt occurred during normal operation or during an Idle. For example, an instruction that activates Idle can also set one or both flag bits. When Idle is terminated by an interrupt, the interrupt service routine can examine the flag bits. The other way of terminating the Idle mode is with a hardware reset. Since the clock oscillator is still running, the hardware reset needs to be held active for only two machine cycles (24 oscillator periods) to complete the reset.



## 80C51 Family

## 80C51 family hardware description

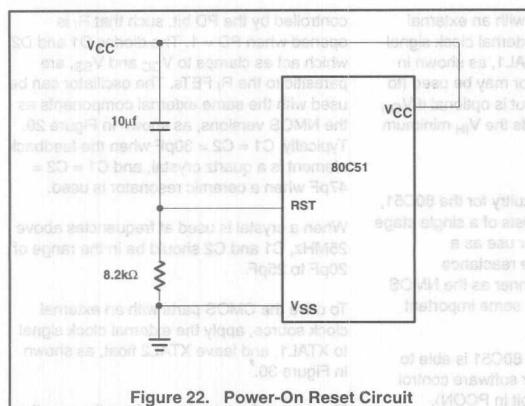


Figure 22. Power-On Reset Circuit

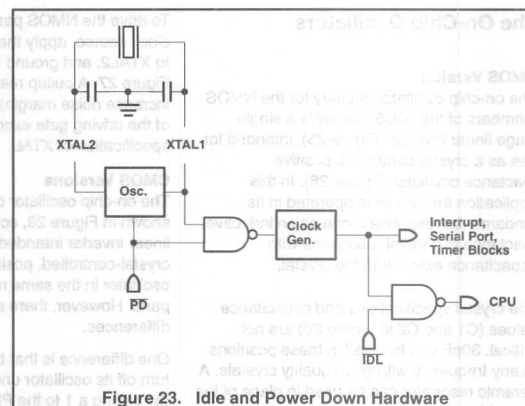


Figure 23. Idle and Power Down Hardware

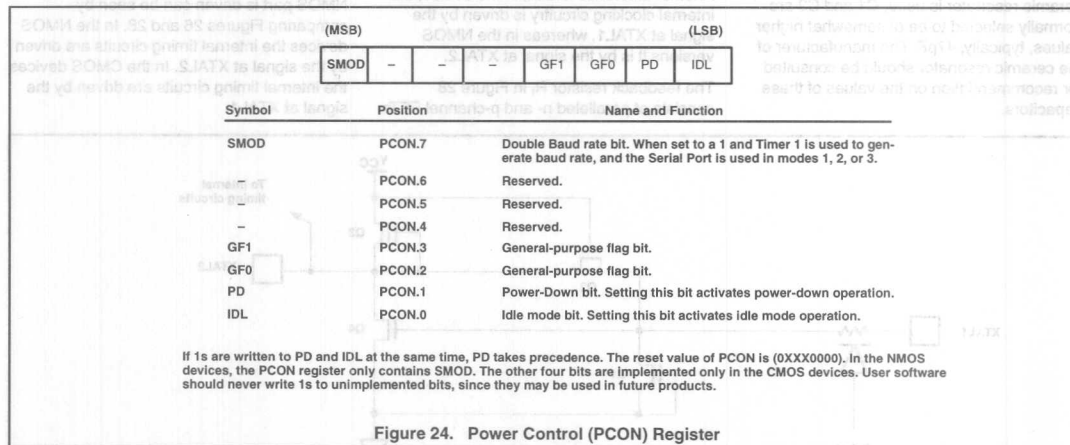


Figure 24. Power Control (PCON) Register

The signal at the RST pin clears the IDL bit directly and asynchronously. At this time the CPU resumes program execution from where it left off; that is, at the instruction following the one that invoked the Idle Mode. As shown in Figure 21, two or three machine cycles of program execution may take place before the internal reset algorithm takes control. On-chip hardware inhibits access to the internal RAM during this time, but access to the port pins is not inhibited, so, the insertion of 3 NOP instructions is recommended following the instruction that invokes idle mode. To eliminate the possibility of unexpected outputs at the port pins, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external Data RAM.

### Power-Down Mode

An instruction that sets PCON.1 causes that to be the last instruction executed before

going into the Power Down mode. In the Power Down mode, the on-chip oscillator is stopped. With the clock frozen, all functions are stopped, the contents of the on-chip RAM and Special Function Registers are maintained. The port pins output the values held by their respective SFRs. The ALE and PSEN output are held low.

The only exit from Power Down is a hardware reset. Reset redefines all the SFRs, but does not change the on-chip RAM.

In the Power Down mode of operation,  $V_{CC}$  can be reduced to as low as 2V. Care must be taken, however, to ensure that  $V_{CC}$  is not reduced before the Power Down mode is invoked, and that  $V_{CC}$  is restored to its normal operating level, before the Power Down mode is terminated. The reset that terminates Power Down also frees the oscillator. The reset should not be activated before  $V_{CC}$  is restored to its normal operating

level, and must be held active long enough to allow the oscillator to restart and stabilize (normally less than 10ms).

### ONCE Mode

The ONCE ("on-circuit emulation") mode facilitates testing and debugging of systems using the device without the device having to be removed from the circuit. The ONCE mode is invoked by:

1. Pull ALE low while the device is in reset and PSEN is high;
2. Hold ALE low as RST is deactivated.

While the device is in the ONCE mode, the Port 0 pins go into a float state, and the other port pins and ALE and PSEN are weakly pulled high. The oscillator circuit remains active. While the device is in this mode, an emulator or test CPU can be used to drive the circuit. Normal operation is restored after a normal reset is applied.



## 80C51 Family

## 80C51 family hardware description

## The On-Chip Oscillators

## NMOS Version

The on-chip oscillator circuitry for the NMOS members of the 80C51 family is a single stage linear inverter (Figure 25), intended for use as a crystal-controlled, positive reactance oscillator (Figure 26). In this application the crystal is operated in its fundamental response mode as an inductive reactance in parallel resonance with capacitance external to the crystal.

The crystal specifications and capacitance values (C1 and C2 in Figure 26) are not critical. 30pF can be used in these positions at any frequency with good quality crystals. A ceramic resonator can be used in place of the crystal in cost-sensitive applications. When a ceramic resonator is used, C1 and C2 are normally selected to be of somewhat higher values, typically, 47pF. The manufacturer of the ceramic resonator should be consulted for recommendation on the values of these capacitors.

To drive the NMOS parts with an external clock source, apply the external clock signal to XTAL2, and ground XTAL1, as shown in Figure 27. A pullup resistor may be used (to increase noise margin), but is optional if  $V_{OH}$  of the driving gate exceeds the  $V_{IH}$  minimum specification of XTAL.

## CMOS Versions

The on-chip oscillator circuitry for the 80C51, shown in Figure 28, consists of a single stage linear inverter intended for use as a crystal-controlled, positive reactance oscillator in the same manner as the NMOS parts. However, there are some important differences.

One difference is that the 80C51 is able to turn off its oscillator under software control (by writing a 1 to the PD bit in PCON). Another difference is that, in the 80C51, the internal clocking circuitry is driven by the signal at XTAL1, whereas in the NMOS versions it is by the signal at XTAL2.

The feedback resistor  $R_f$  in Figure 28 consists of paralleled n- and p-channel FETs

controlled by the PD bit, such that  $R_f$  is opened when  $PD = 1$ . The diodes D1 and D2, which act as clamps to  $V_{CC}$  and  $V_{SS}$ , are parasitic to the  $R_f$  FETs. The oscillator can be used with the same external components as the NMOS versions, as shown in Figure 29. Typically,  $C1 = C2 = 30pF$  when the feedback element is a quartz crystal, and  $C1 = C2 = 47pF$  when a ceramic resonator is used.

When a crystal is used at frequencies above 25MHz, C1 and C2 should be in the range of 20pF to 25pF.

To drive the CMOS parts with an external clock source, apply the external clock signal to XTAL1, and leave XTAL2 float, as shown in Figure 30.

The reason for this change from the way the NMOS part is driven can be seen by comparing Figures 26 and 28. In the NMOS devices the internal timing circuits are driven by the signal at XTAL2. In the CMOS devices the internal timing circuits are driven by the signal at XTAL1:

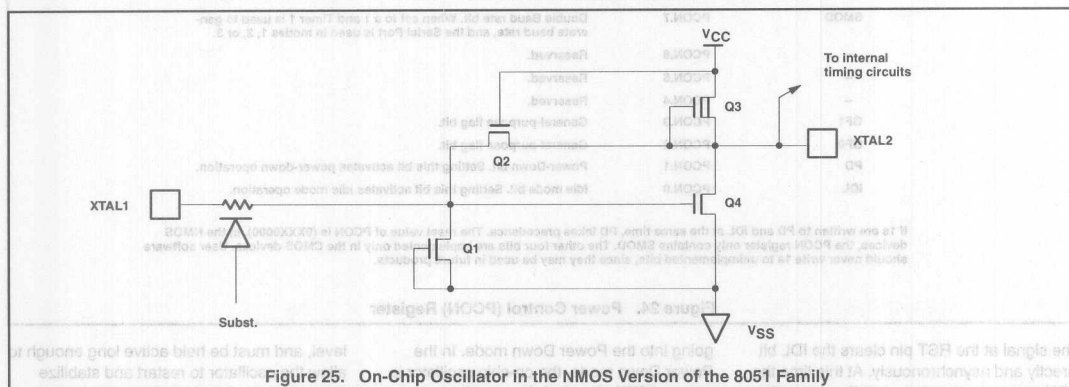


Figure 25. On-Chip Oscillator in the NMOS Version of the 8051 Family

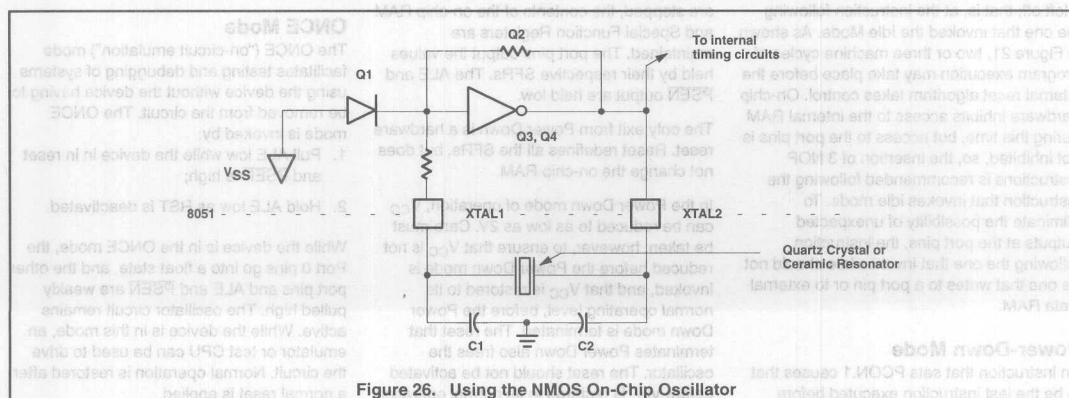


Figure 26. Using the NMOS On-Chip Oscillator

## 80C51 Family

## 80C51 family hardware description

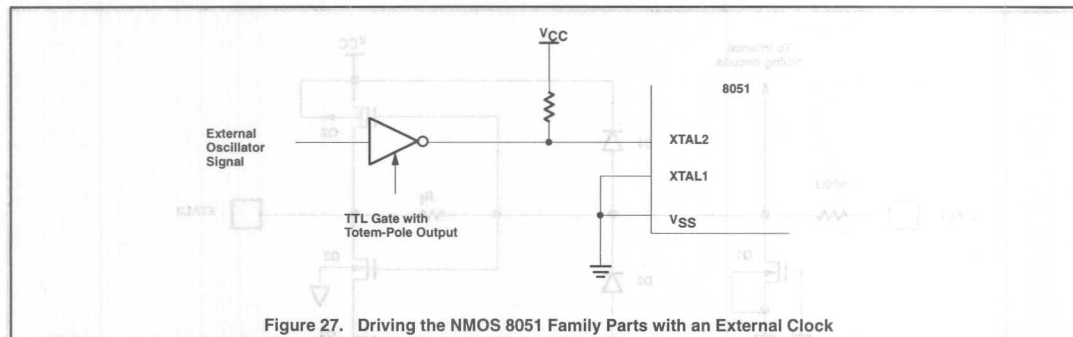


Figure 27. Driving the NMOS 8051 Family Parts with an External Clock

## Internal Timing

Figures 31 through 34 show when the various strobe and port signals are clocked internally. The figures do not show rise and fall times of the signals, nor do they show propagation delays between the XTAL2 signal and events at other pins.

Rise and fall times are dependent on the external loading that each pin must drive. They are often taken to be something in the neighborhood of 10ns, measured between 0.8V and 2.0V.

Propagation delays are different for different pins. For a given pin they vary with pin loading, temperature,  $V_{CC}$ , and manufacturing lot. If the XTAL2 waveform is taken as the timing reference, prop delays may vary up to  $\pm 200\%$ .

The AC Timings section of the data sheets do not reference any timing to the XTAL2 waveform. Rather, they relate the critical edges of control and input signals to each other. The timings published in the data sheets include the effects of propagation delays under the specified test conditions.

## 80C51 Pin Descriptions

**ALE/PROG:** Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. ALE is emitted at a constant rate of 1/6 of the oscillator frequency, for external timing or clocking purposes, even when there are no accesses to external memory. (However, one ALE pulse is skipped during each access to

external Data Memory.) This pin is also the program pulse input (PROG) during EPROM programming.

**PSEN:** Program Store Enable is the read strobe to external Program Memory. When the device is executing out of external Program Memory, PSEN is activated twice each machine cycle (except that two PSEN activations are skipped during accesses to external Data Memory). PSEN is not activated when the device is executing out of internal Program Memory.

**EA/ $V_{PP}$ :** When  $\overline{EA}$  is held high the CPU executes out of internal Program Memory (unless the Program Counter exceeds 0FFFh in the 80C51). Holding  $\overline{EA}$  low forces the CPU to execute out of external memory regardless of the Program Counter value. In the 80C31,  $\overline{EA}$  must be externally wired low. In the EPROM devices, this pin also receives the programming supply voltage ( $V_{PP}$ ) during EPROM programming.

**XTAL1:** Input to the inverting oscillator amplifier.

**XTAL2:** Output from the inverting oscillator amplifier.

**Port 0:** Port 0 is an 8-bit open drain bidirectional port. As an open drain output port, it can sink eight LS TTL loads. Port 0 pins that have 1s written to them float, and in that state will function as high impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external memory. In this

application it uses strong internal pullups when emitting 1s. Port 0 emits code bytes during program verification. In this application, external pullups are required.

**Port 1:** Port 1 is an 8-bit bidirectional I/O port with internal pullups. Port 1 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, port 1 pins that are externally being pulled low will source current because of the internal pullups.

**Port 2:** Port 2 is an 8-bit bidirectional I/O port with internal pullups. Port 2 emits the high-order address byte during accesses to external memory that use 16-bit addresses. In this application, it uses the strong internal pullups when emitting 1s.

**Port 3:** Port 3 is an 8-bit bidirectional I/O port with internal pullups. It also serves the functions of various special features of the 80C51 Family as follows:

Port Pin	Alternate Function
P3.0	RxD (serial input port)
P3.1	TxD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	WR (external data memory write strobe)
P3.7	RD (external data memory read strobe)

$V_{CC}$ : Supply voltage

$V_{SS}$ : Circuit ground potential

# 80C51 Family

## 80C51 family hardware description

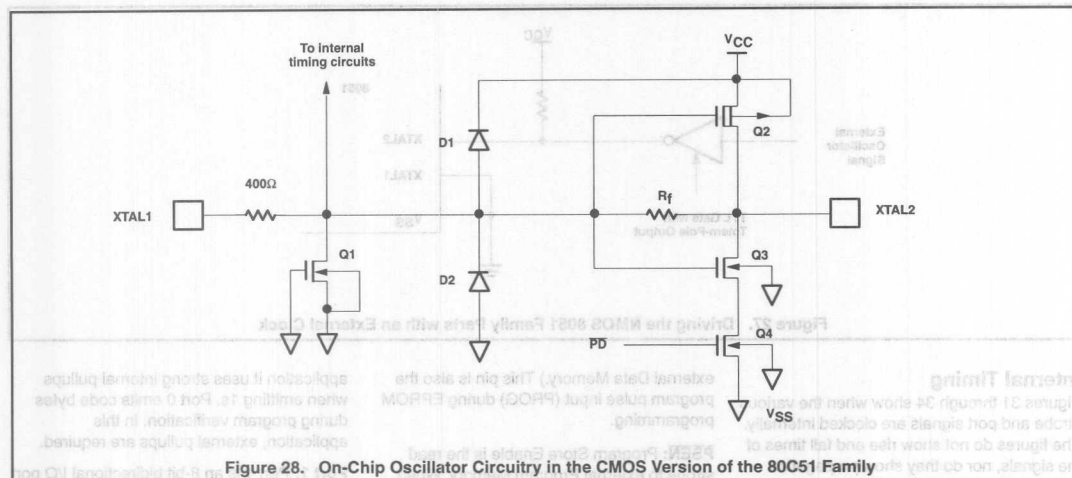


Figure 28. On-Chip Oscillator Circuitry in the CMOS Version of the 80C51 Family

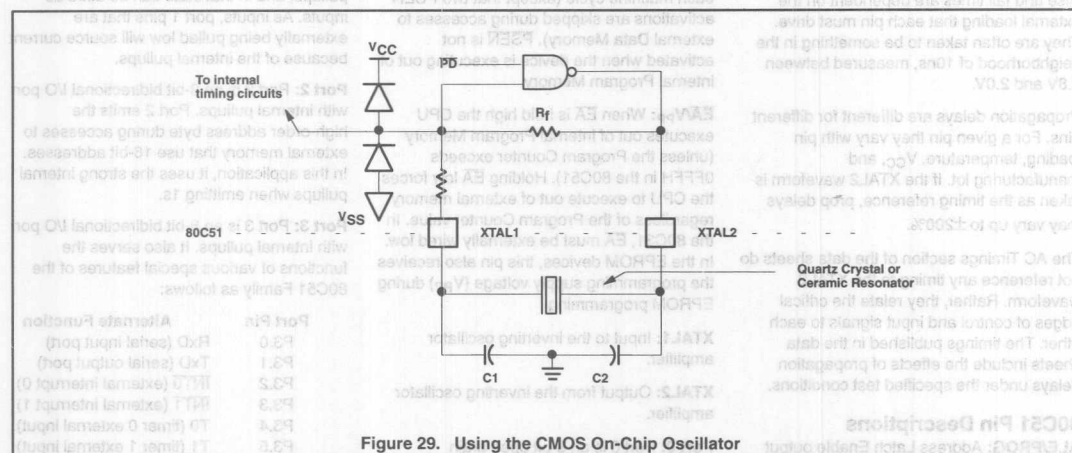


Figure 29. Using the CMOS On-Chip Oscillator

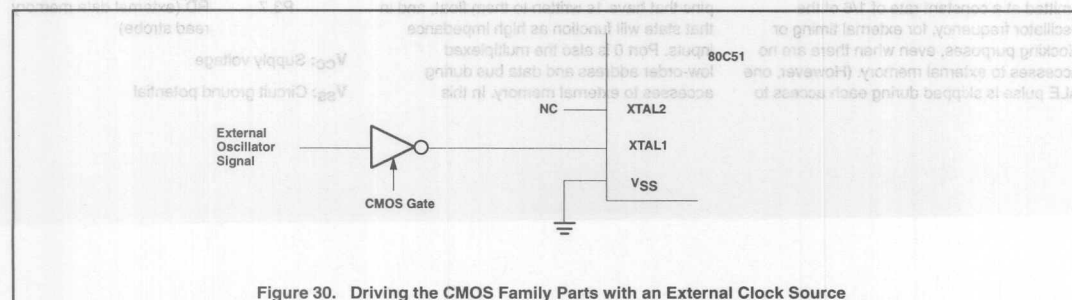
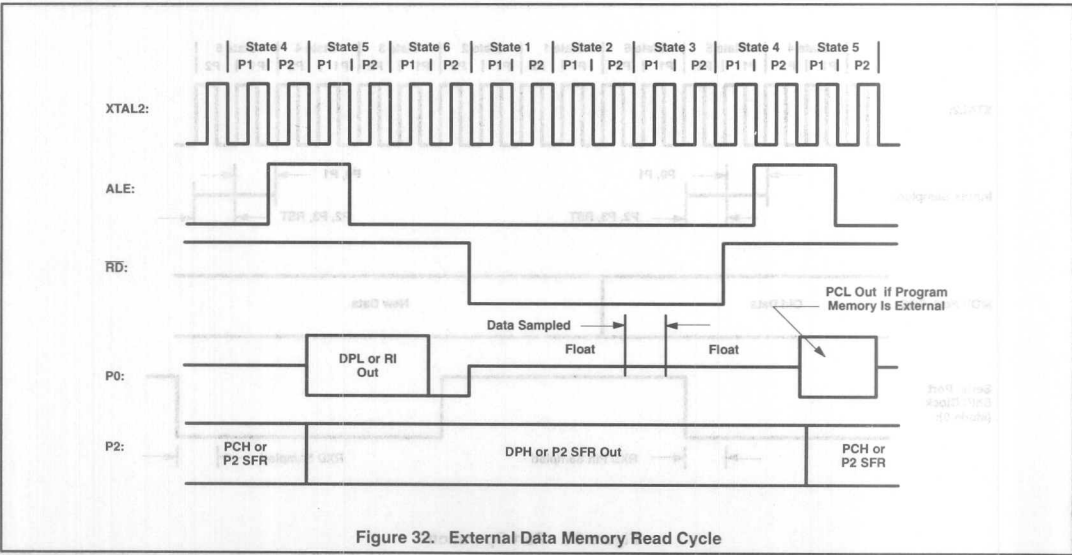
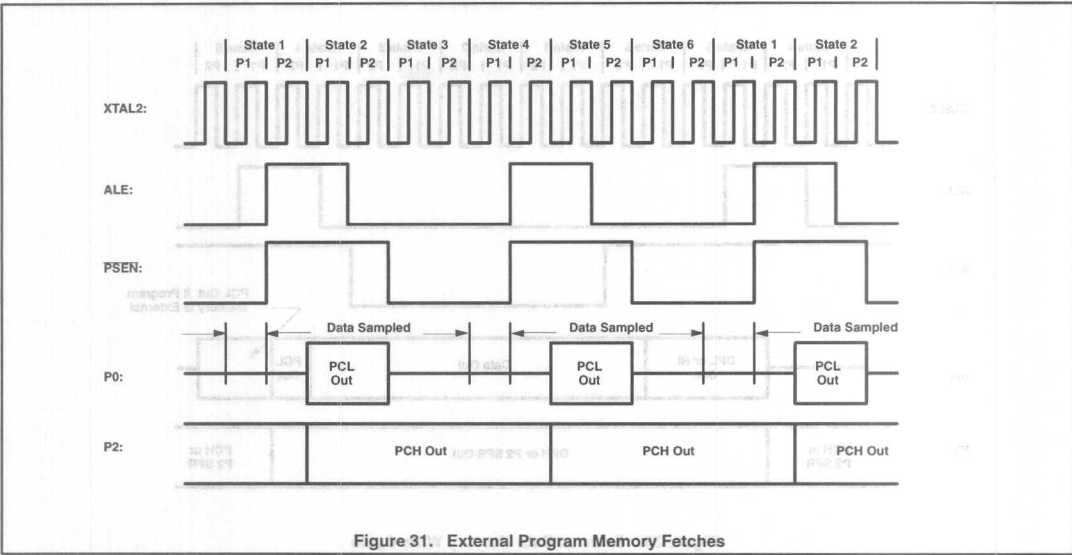
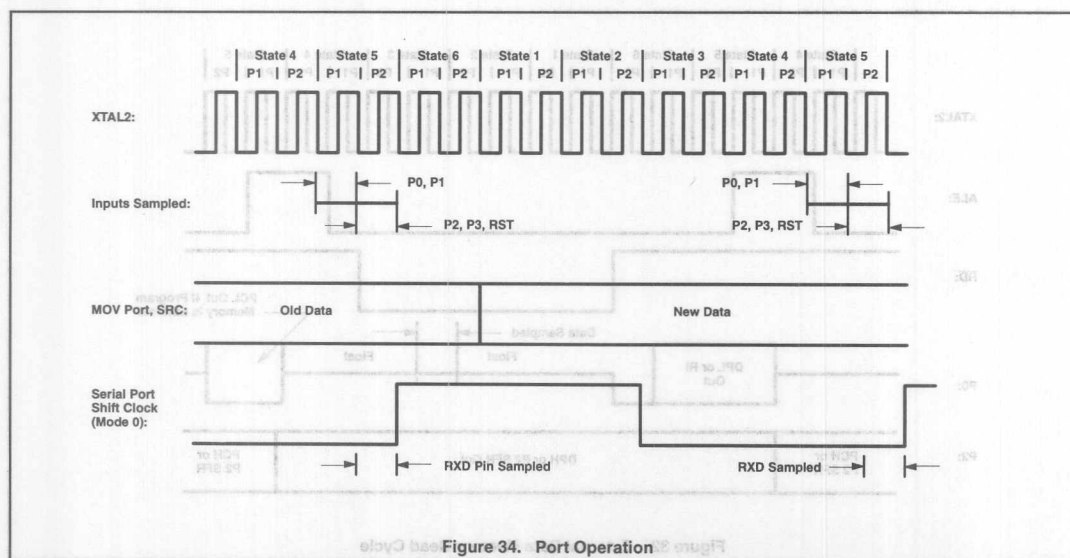
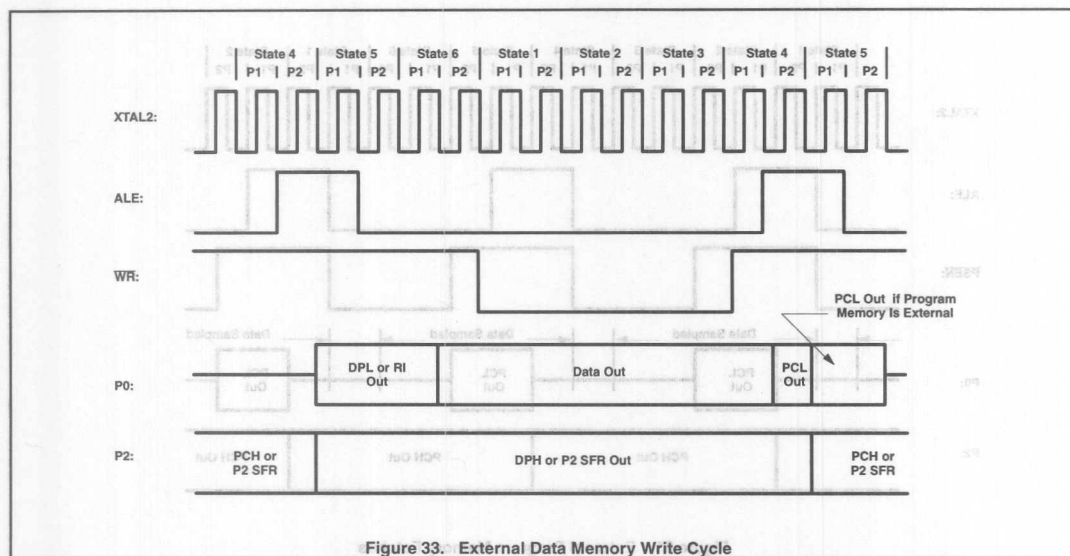


Figure 30. Driving the CMOS Family Parts with an External Clock Source



## 80C51 Family

## 80C51 family hardware description



# 80C51 family programmer's guide and instruction set

## 80C51 Family

### PROGRAMMER'S GUIDE AND INSTRUCTION SET

#### Memory Organization

##### Program Memory

The 80C51 has separate address spaces for program and data memory. The Program memory can be up to 64k bytes long. The lower 4k can reside on-chip. Figure 1 shows a map of the 80C51 program memory.

The 80C51 can address up to 64k bytes of data memory to the chip. The MOVX instruction is used to access the external data memory.

The 80C51 has 128 bytes of on-chip RAM, plus a number of Special Function Registers (SFRs). The lower 128 bytes of RAM can be accessed either by direct addressing (MOV data addr) or by indirect addressing (MOV @Ri). Figure 2 shows the Data Memory organization.

##### Direct and Indirect Address Area

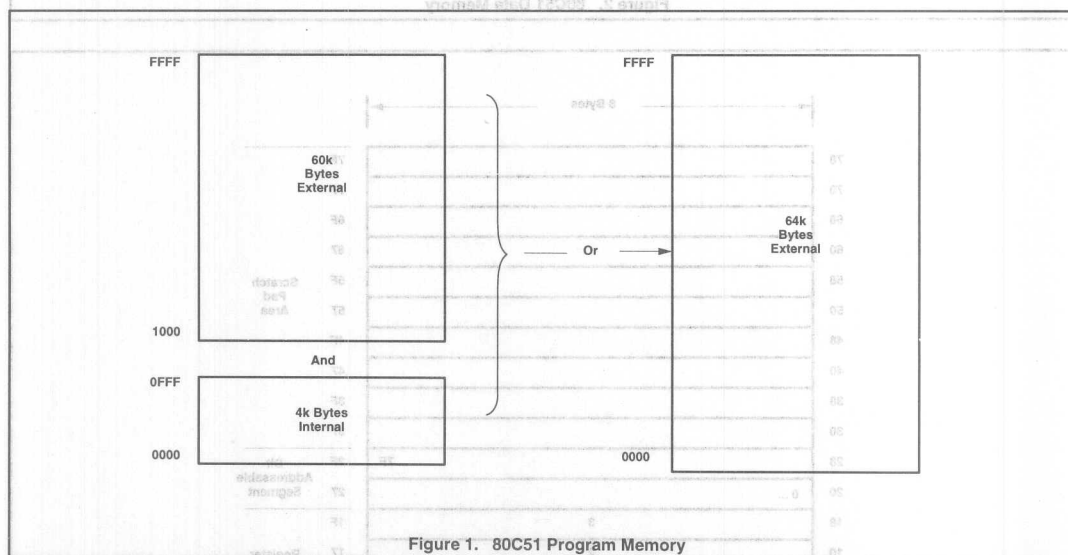
The 128 bytes of RAM which can be accessed by both direct and indirect addressing can be divided into three segments as listed below and shown in Figure 3.

1. Register Banks 0-3: Locations 0 through 1FH (32 bytes). The device after reset defaults to register bank 0. To use the other register banks, the user must select them in software. Each register bank contains eight 1-byte registers 0 through 7. Reset initializes the stack pointer to location 07H, and it is incremented once to start from location 08H, which is the first register (R0) of the second register bank. Thus, in order to use more than one register bank, the SP should be initialized to a different location of the RAM where it is not used for data storage (i.e., the higher part of the RAM).
2. Bit Addressable Area: 16 bytes have been assigned for this segment, 20H-2FH.

Each one of the 128 bits of this segment can be directly addressed (0-7FH). The bits can be referred to in two ways, both of which are acceptable by most assemblers. One way is to refer to their address (i.e., 0-7FH). The other way is with reference to bytes 20H to 2FH. Thus, bits 0-7 can also be referred to as bits 20.0-20.7, and bits 8-FH are the same as 21.0-21.7, and so on. Each of the 16 bytes in this segment can also be addressed as a byte.

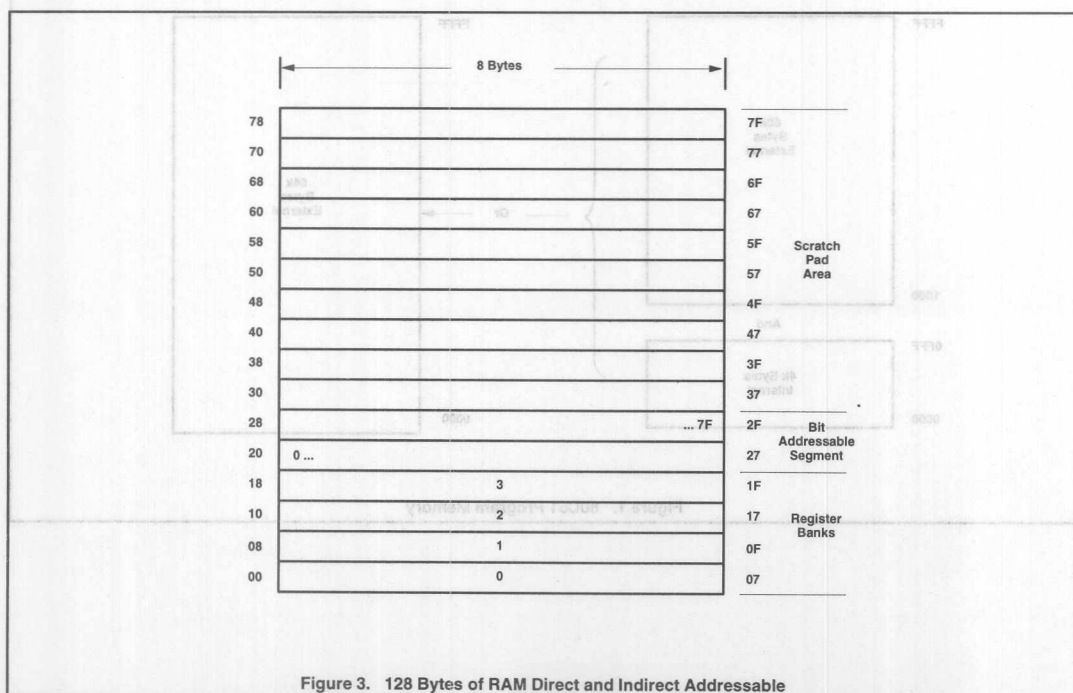
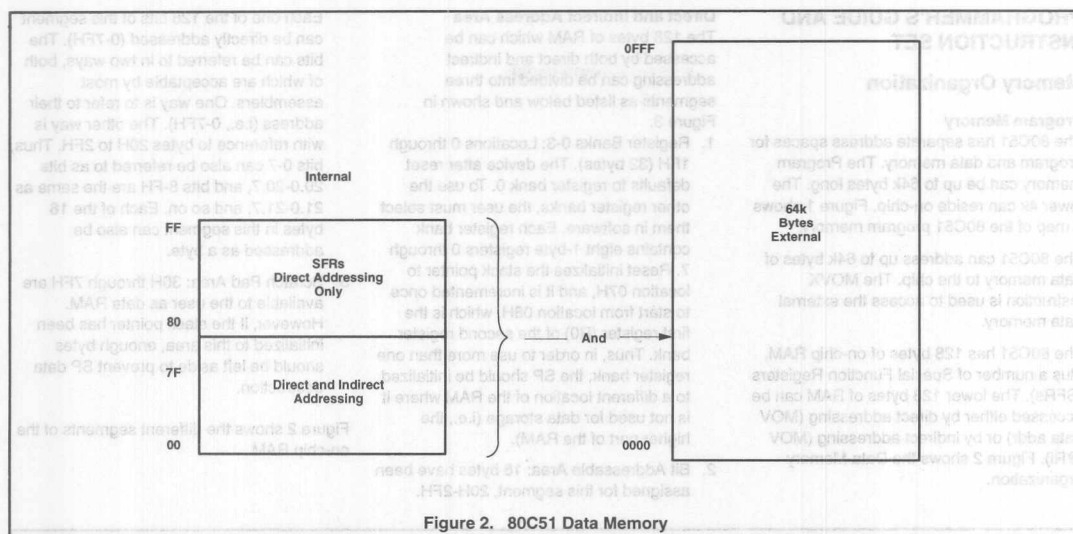
3. Scratch Pad Area: 30H through 7FH are available to the user as data RAM. However, if the stack pointer has been initialized to this area, enough bytes should be left aside to prevent SP data destruction.

Figure 2 shows the different segments of the on-chip RAM.





## 80C51 Family

80C51 family programmer's guide  
and instruction set

## 80C51 family

80C51 family programmer's guide  
and instruction set

Table 1. 80C51 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB							LSB	
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
DPTR	Data pointer (2 bytes)										
DPH	Data pointer high	83H									00H
DPL	Data pointer low	82H									00H
			AF	AE	AD	AC	AB	AA	A9	A8	
IE*	Interrupt enable	A8H	EA	—	—	ES	ET1	EX1	ET0	EX0	0x000000B
			BF	BE	BD	BC	BB	BA	B9	B8	
IP*	Interrupt priority	B8H	—	—	—	PS	PT1	PX1	PT0	PX0	xx000000B
			87	86	85	84	83	82	81	80	
P0*	Port 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
			97	96	95	94	93	92	91	90	
P1*	Port 1	90H	—	—	—	—	—	—	T2EX	T2	FFH
			A7	A6	A5	A4	A3	A2	A1	A0	
P2*	Port 2	A0H	A15	A14	A13	A12	A11	A10	A9	A8	FFH
			B7	B6	B5	B4	B3	B2	B1	B0	
P3*	Port 3	B0H	RD	WR	T1	T0	INT1	INT0	TxD	RxD	FFH
PCON <sup>1</sup>	Power control	87H	SMOD	—	—	—	GF1	GF0	PD	IDL	0xxxxxxxB
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program status word	D0H	CY	AC	F0	RS1	RS0	OV	—	P	00H
SBUF	Serial data buffer	99H									xxxxxxxB
			9F	9E	9D	9C	9B	9A	99	98	
SCON*	Serial controller	98H	SM0	SM1	SM2	REN	TB8	RB8	T1	RI	00H
SP	Stack pointer	81H									07H
			8F	8E	8D	8C	8B	8A	89	88	
TCON*	Timer control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	
TH0	Timer high 0	8CH									00H
TH1	Timer high 1	8DH									00H
TL0	Timer low 0	8AH									00H
TL1	Timer low 1	8BH									00H
TMOD	Timer mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H

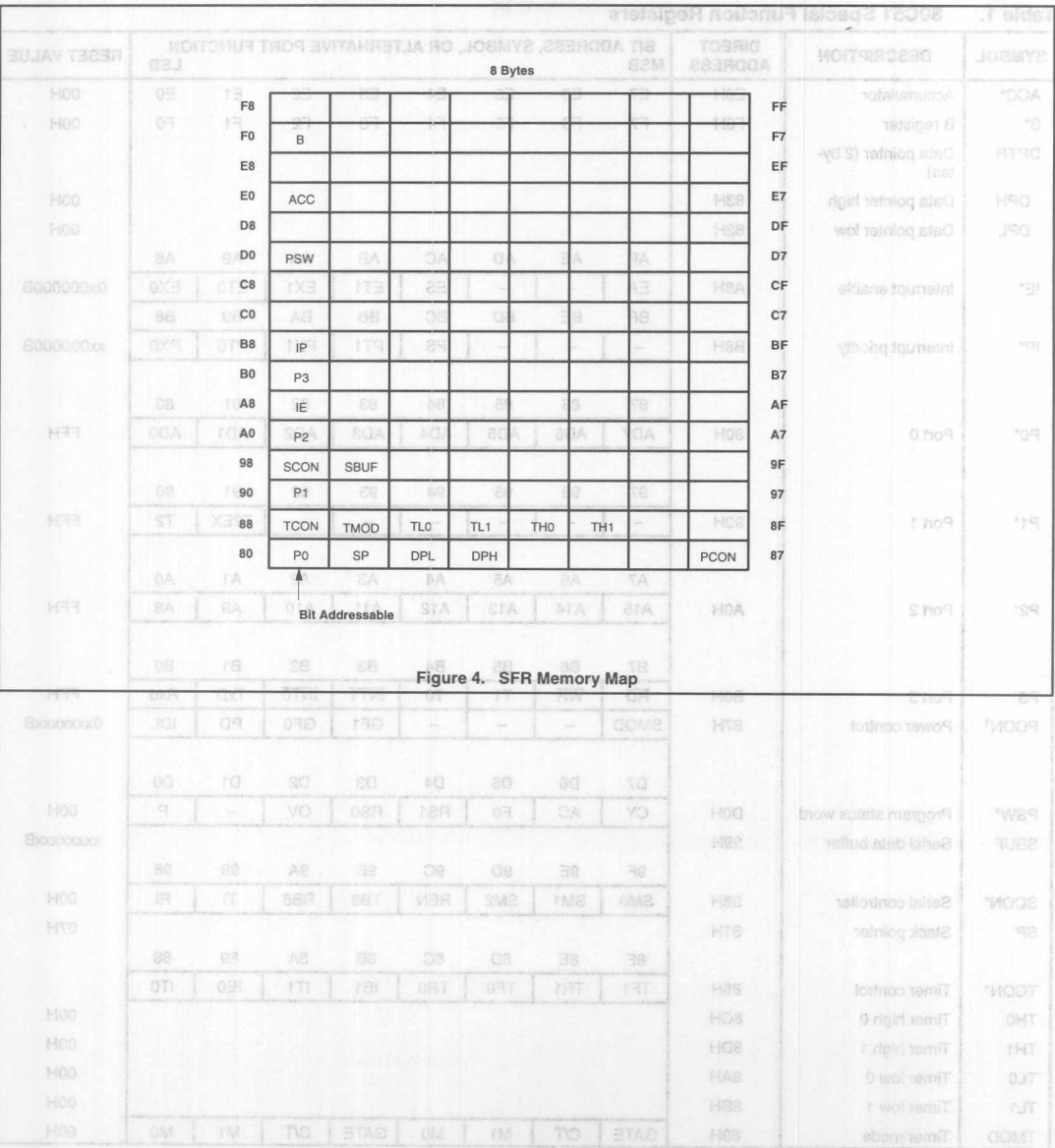
## NOTES:

\* Bit addressable

1. Bits GF1, GF0, PD, and IDL of the PCON register are not implemented on the NMOS 8051/8031.

80C51 Family

80C51 family programmer's guide  
and instruction set



## 80C51 Family

80C51 family programmer's guide  
and instruction set

Those SFRs that have their bits assigned for various functions are listed in this section. A brief description of each bit is provided for quick reference. For more detailed information refer to the Architecture Chapter of this book.

**PSW: PROGRAM STATUS WORD. BIT ADDRESSABLE.**

CY	AC	F0	RS1	RS0	OV	—	P
----	----	----	-----	-----	----	---	---

CY	PSW.7	Carry Flag.					
AC	PSW.6	Auxiliary Carry Flag.					
F0	PSW.5	Flag 0 available to the user for general purpose.					
RS1	PSW.4	Register Bank selector bit 1 (SEE NOTE 1).					
RS0	PSW.3	Register Bank selector bit 0 (SEE NOTE 1).					
OV	PSW.2	Overflow Flag.					
—	PSW.1	Usable as a general purpose flag.					
P	PSW.0	Parity flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of '1' bus in the accumulator.					

**NOTE:**

1. The value presented by RS0 and RS1 selects the corresponding register bank.

RS1	RS0	REGISTER BANK	ADDRESS
0	0	0	00H-07H
0	1	1	08H-0FH
1	0	2	10H-17H
1	1	3	18H-1FH

**PCON: POWER CONTROL REGISTER. NOT BIT ADDRESSABLE.**

SMOD	—	—	—	GF1	GF0	PD	IDL
------	---	---	---	-----	-----	----	-----

SMOD Double baud rate bit. If Timer 1 is used to generate baud rate and SMOD = 1, the baud rate is doubled when the Serial Port is used in modes 1, 2, or 3.

— Not implemented, reserved for future use.\*

— Not implemented reserved for future use.\*

— Not implemented reserved for future use.\*

GF1 General purpose flag bit.

GF0 General purpose flag bit.

PD Power Down Bit. Setting this bit activates Power Down operation in the 80C51. (Available only in CMOS.)

IDL Idle mode bit. Setting this bit activates Idle Mode operation in the 80C51. (Available only in CMOS.)

If 1s are written to PD and IDL at the same time, PD takes precedence.

\* User software should not write 1s to reserved bits. These bits may be used in future 8051 products to invoke new features.

80C51 Family

80C51 family programmer's guide  
and instruction set

INTERRUPTS:

To use any of the interrupts in the 80C51 Family, the following three steps must be taken.

1. Set the EA (enable all) bit in the IE register to 1.
2. Set the corresponding individual interrupt enable bit in the IE register to 1.
3. Begin the interrupt service routine at the corresponding Vector Address of that interrupt. See Table below.

INTERRUPT SOURCE	VECTOR ADDRESS
IE0	0003H
TF0	000BH
IE1	0013H
TF1	001BH
RI & TI	0023H

In addition, for external interrupts, pins INT0 and INT1 (P3.2 and P3.3) must be set to 1, and depending on whether the interrupt is to be level or transition activated, bits IT0 or IT1 in the TCON register may need to be set to 1.

ITx = 0 level activated

ITx = 1 transition activated

IE: INTERRUPT ENABLE REGISTER. BIT ADDRESSABLE.

If the bit is 0, the corresponding interrupt is disabled. If the bit is 1, the corresponding interrupt is enabled.

EA	—	—	ES	ET1	EX1	ET0	EX0
----	---	---	----	-----	-----	-----	-----

EA	IE.7	Disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	Not implemented, reserved for future use.*
—	IE.5	Not implemented, reserved for future use.*
ES	IE.4	Enable or disable the serial port interrupt.
ET1	IE.3	Enable or disable the Timer 1 overflow interrupt.
EX1	IE.2	Enable or disable External Interrupt 1.
ET0	IE.1	Enable or disable the Timer 0 overflow interrupt.
EX0	IE.0	Enable or disable External Interrupt 0.

\* User software should not write 1s to reserved bits. These bits may be used in future 80C51 products to invoke new features.

## 80C51 Family

80C51 family programmer's guide  
and instruction set**ASSIGNING HIGHER PRIORITY TO ONE OR MORE INTERRUPTS:**

In order to assign higher priority to an interrupt the corresponding bit in the IP register must be set to 1.

Remember that while an interrupt service is in progress, it cannot be interrupted by a lower or same level interrupt.

**PRIORITY WITHIN LEVEL:**

Priority within level is only to resolve simultaneous requests of the same priority level.

From high to low, interrupt sources are listed below:

IE0

TF0

IE1

TF1

RI or TI

**IP: INTERRUPT PRIORITY REGISTER. BIT ADDRESSABLE.**

If the bit is 0, the corresponding interrupt has a lower priority and if the bit is 1 the corresponding interrupt has a higher priority.

—	—	—	PS	PT1	PX1	PT0	PX0

—	IP.7	Not implemented, reserved for future use.*
—	IP.6	Not implemented, reserved for future use.*
—	IP.5	Not implemented, reserved for future use.*
PS	IP.4	Defines the Serial Port interrupt priority level.
PT1	IP.3	Defines the Timer 1 interrupt priority level.
PX1	IP.2	Defines External Interrupt 1 priority level.
PT0	IP.1	Defines the Timer 0 interrupt priority level.
PX0	IP.0	Defines the External Interrupt 0 priority level.

\* User software should not write 1s to reserved bits. These bits may be used in future 80C51 products to invoke new features.

Mode	MD	Operating Mode
0	0	13-bit Timer (8048 compatible)
1	1	16-bit Timer/Counter
2	0	8-bit Auto-Reload Timer/Counter
3	1	Timer 0 (T0) is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. This is an 8-bit timer and is controlled by Timer 1 control bits. (Timer 1) Timer/Counter 1 stopped.



## 80C51 Family

80C51 family programmer's guide  
and instruction set**TCON: TIMER/COUNTER CONTROL REGISTER. BIT ADDRESSABLE.**

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
TF1	TCON.7	Timer 1 overflow flag. Set by hardware when the Timer/Counter 1 overflows. Cleared by hardware as processor vectors to the interrupt service routine.					
TR1	TCON.6	Timer 1 run control bit. Set/cleared by software to turn Timer/Counter 1 ON/OFF.					
TF0	TCON.5	Timer 0 overflow flag. Set by hardware when the Timer/Counter 0 overflows. Cleared by hardware as processor vectors to the service routine.					
TR0	TCON.4	Timer 0 run control bit. Set/cleared by software to turn Timer/Counter 0 ON/OFF.					
IE1	TCON.3	External Interrupt 1 edge flag. Set by hardware when External Interrupt edge is detected. Cleared by hardware when interrupt is processed.					
IT1	TCON.2	Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.					
IE0	TCON.1	External Interrupt 0 edge flag. Set by hardware when External Interrupt edge detected. Cleared by hardware when interrupt is processed.					
IT0	TCON.0	Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.					

**TMOD: TIMER/COUNTER MODE CONTROL REGISTER. NOT BIT ADDRESSABLE.**

GATE	C/T	M1	M0	GATE	C/T	M1	M0
Timer 1				Timer 0			

GATE	When TRx (in TCON) is set and GATE = 1, TIMER/COUNTERx will run only while INTx pin is high (hardware control). When GATE = 0, TIMER/COUNTERx will run only while TRx = 1 (software control).
C/T	Timer or Counter selector. Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from Tx input pin).
M1	Mode selector bit. (NOTE 1)
M0	Mode selector bit. (NOTE 1)

**NOTE 1:**

M1	M0	Operating Mode
0	0	0 13-bit Timer (8048 compatible)
0	1	1 16-bit Timer/Counter
1	0	2 8-bit Auto-Reload Timer/Counter
1	1	3 (Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standart Timer 0 control bits. TH0 is an 8-bit Timer and is controlled by Timer 1 control bits.
1	1	3 (Timer 1) Timer/Counter 1 stopped.

## 80C51 Family

80C51 family programmer's guide  
and instruction set

## TIMER SET-UP

Tables 2 through 5 give some values for TMOD which can be used to set up Timer 0 in different modes.

It is assumed that only one timer is being used at a time. If it is desired to run Timers 0 and 1 simultaneously, in any mode, the value in TMOD for Timer 0 must be ORed with the value shown for Timer 1 (Tables 5 and 6).

For example, if it is desired to run Timer 0 in mode 1 GATE (external control), and Timer 1 in mode 2 COUNTER, then the value that must be loaded into TMOD is 69H (09H from Table 3 ORed with 60H from Table 6).

Moreover, it is assumed that the user, at this point, is not ready to turn the timers on and will do that at a different point in the program by setting bit TRx (in TCON) to 1.

## TIMER/COUNTER 0

Table 2. As a Timer:

MODE	TIMER 0 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	00H	08H
1	16-bit Timer	01H	09H
2	8-bit Auto-Reload	02H	0AH
3	Two 8-bit Timers	03H	0BH

Table 3. As a Counter:

MODE	COUNTER 0 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	04H	0CH
1	16-bit Timer	05H	0DH
2	8-bit Auto-Reload	06H	0EH
3	One 8-bit Counter	07H	0FH

## NOTES:

1. The timer is turned ON/OFF by setting/clearing bit TR0 in the software.
2. The Timer is turned ON/OFF by the 1-to-0 transition on INT0 (P3.2) when TR0 = 1 (hardware control).

80C51 Family

80C51 family programmer's guide  
and instruction set

TIMER/COUNTER 1

Table 4. As a Timer:

MODE	TIMER 1 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	00H	80H
1	16-bit Timer	10H	90H
2	8-bit Auto-Reload	20H	A0H
3	Does not run	30H	B0H

Table 5. As a Counter:

MODE	COUNTER 1 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	40H	C0H
1	16-bit Timer	50H	D0H
2	8-bit Auto-Reload	60H	E0H
3	Not available	—	—

NOTES:

1. The timer is turned ON/OFF by setting/clearing bit TR1 in the software.
2. The Timer is turned ON/OFF by the 1-to-0 transition on INT1 (P3.2) when TR1 = 1 (hardware control).

MODE	COUNTER 0 FUNCTION	INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	40H	C0H
1	16-bit Timer	50H	D0H
2	8-bit Auto-Reload	60H	E0H
3	One 8-bit Counter	70H	F0H

NOTES:

1. The timer is turned ON/OFF by setting/clearing bit TR0 in the software.
2. The Timer is turned ON/OFF by the 1-to-0 transition on INT0 (P3.2) when TR0 = 1 (hardware control).

80C51 Family

80C51 family programmer's guide  
and instruction set

SCON: SERIAL PORT CONTROL REGISTER. BIT ADDRESSABLE

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
SM0	SCON.7	Serial Port mode specifier. (NOTE 1)					
SM1	SCON.6	Serial Port mode specifier. (NOTE 1)					
SM2	SCON.5	Enables the multiprocessor communication feature in modes 2 & 3. In mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0. (See Table 6.)					
REN	SCON.4	Set/Cleared by software to Enable/Disable reception.					
TB8	SCON.3	The 9th bit that will be transmitted in modes 2 & 3. Set/Cleared by software.					
RB8	SCON.2	In modes 2 & 3, is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.					
TI	SCON.1	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes. Must be cleared by software.					
RI	SCON.0	Receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bit time in the other modes (except see SM2). Must be cleared by software.					

NOTE 1:

SM0	SM1	Mode	Description	Baud Rate
0	0	0	Shift Register	F <sub>osc</sub> /12
0	1	1	8-bit UART	Variable
1	0	2	9-bit UART	F <sub>osc</sub> /64 or F <sub>osc</sub> /32
1	1	3	9-bit UART	Variable

SERIAL PORT SET-UP:

Table 6.

MODE	SCON	SM2 VARIATION
0 1 2 3	10H 50H 90H D0H	Single Processor Environment (SM2 = 0)
0 1 2 3	NA 70H B0H F0H	Multiprocessor Environment (SM2 = 1)

GENERATING BAUD RATES

Serial Port in Mode 0:

Mode 0 has a fixed baud rate which is 1/12 of the oscillator frequency. To run the serial port in this mode none of the Timer/Counters need to be set up. Only the SCON register needs to be defined.

Baud Rate =  $\frac{\text{Osc Freq}}{12}$

Serial Port in Mode 1:

Mode 1 has a variable baud rate. The baud rate is generated by Timer 1.

80C51 Family

80C51 family programmer's guide  
and instruction set

USING TIMER/COUNTER 1 TO GENERATE BAUD RATES:

For this purpose, Timer 1 is used in mode 2 (Auto-Reload). Refer to Timer Setup section of this chapter.

$$\text{Baud Rate} = \frac{K \times \text{Osc Freq}}{32 \times 12 \times [256 - (TH1)]}$$

If SMOD = 0, then K = 1.

If SMOD = 1, then K = 2 (SMOD is in the PCON register).

Most of the time the user knows the baud rate and needs to know the reload value for TH1.

$$TH1 = 256 - \frac{K \times \text{Osc Freq}}{384 \times \text{baud rate}}$$

TH1 must be an integer value. Rounding off TH1 to the nearest integer may not produce the desired baud rate. In this case, the user may have to choose another crystal frequency.

Since the PCON register is not bit addressable, one way to set the bit is logical ORing the PCON register (i.e., ORL PCON,#80H). The address of PCON is 87H.

SERIAL PORT IN MODE 2:

The baud rate is fixed in this mode and is 1/32 or 1/64 of the oscillator frequency, depending on the value of the SMOD bit in the PCON register.

In this mode none of the Timers are used and the clock comes from the internal phase 2 clock.

SMOD = 1, Baud Rate = 1/32 Osc Freq.

SMOD = 0, Baud Rate = 1/64 Osc Freq.

To set the SMOD bit: ORL PCON,#80H. The address of PCON is 87H.

SERIAL PORT IN MODE 3:

The baud rate in mode 3 is variable and sets up exactly the same as in mode 1.

MODE	SCON	SMS VARIATION
0	10H	Single Processor Environment (SM2 = 0)
1	20H	
2	30H	
3	00H	
0	4A	Multiprocessor Environment (SM2 = 1)
1	70H	
2	60H	
3	50H	

## 80C51 FAMILY INSTRUCTION SET

Table 7. 80C51 Instruction Set Summary

Interrupt Response Time: Refer to Hardware Description Chapter.			
Instructions that Affect Flag Settings <sup>(1)</sup>			
Instruction	Flag	Instruction	Flag
	C OV AC		C OV AC
ADD	X X X	CLR C	0
ADDC	X X X	CPL C	X
SUBB	X X X	ANL C,bit	X
MUL	0 X	ANL C,bit	X
DIV	0 X	ANL C,bit	X
DA	X	ORL C,bit	X
RRC	X	MOV C,bit	X
RLC	X	CJNE	X
SETB C	1		

<sup>(1)</sup>Note that operations on SFR byte address 208 or bit addresses 209-215 (i.e., the PSW or bits in the PSW) will also affect flag settings.

**Notes on instruction set and addressing modes:**

Rn Register R7-R0 of the currently selected Register Bank.

direct 8-bit internal data location's address. This could be an Internal Data RAM location (0-127) or a SFR [i.e., I/O port, control register, status register, etc. (128-255)].

@Ri 8-bit internal data RAM location (0-255) addressed indirectly through register R1 or R0.

#data 8-bit constant included in the instruction.

#data 16 16-bit constant included in the instruction.

addr 16 16-bit destination address. Used by LCALL and LJMP. A branch can be anywhere within the 64k-byte Program Memory address space.

addr 11 11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2k-byte page of program memory as the first byte of the following instruction.

rel Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is -128 to +127 bytes relative to first byte of the following instruction.

bit Direct Addressed bit in Internal Data RAM or Special Function Register.

MNEMONIC	DESCRIPTION	BYTE	OSCILLATOR PERIOD
<b>ARITHMETIC OPERATIONS</b>			
ADD A,Rn	Add register to Accumulator	1	12
ADD A,direct	Add direct byte to Accumulator	2	12
ADD A,@Ri	Add indirect RAM to Accumulator	1	12
ADD A,#data	Add immediate data to Accumulator	2	12
ADDC A,Rn	Add register to Accumulator with carry	1	12
ADDC A,direct	Add direct byte to Accumulator with carry	2	12
ADDC A,@Ri	Add indirect RAM to Accumulator with carry	1	12
ADDC A,#data	Add immediate data to ACC with carry	2	12
SUBB A,Rn	Subtract Register from ACC with borrow	1	12
SUBB A,direct	Subtract direct byte from ACC with borrow	2	12
SUBB A,@Ri	Subtract indirect RAM from ACC with borrow	1	12
SUBB A,#data	Subtract immediate data from ACC with borrow	2	12
INC A	Increment Accumulator	1	12
INC Rn	Increment register	1	12

0887 Intel Corporation 1980

All mnemonics copyrighted © Intel Corporation 1980



## 80C51 Family

80C51 family programmer's guide  
and instruction set

Table 7. 80C51 Instruction Set Summary (Continued)

MNEMONIC		DESCRIPTION	BYTE	OSCILLATOR PERIOD
<b>ARITHMETIC OPERATIONS (Continued)</b>				
INC	direct	Increment direct byte	2	12
INC	@Ri	Increment indirect RAM	1	12
DEC	A	Decrement Accumulator	1	12
DEC	Rn	Decrement Register	1	12
DEC	direct	Decrement direct byte	2	12
DEC	@Ri	Decrement indirect RAM	1	12
INC	DPTR	Increment Data Pointer	1	24
MUL	AB	Multiply A and B	1	48
DIV	AB	Divide A by B	1	48
DA	A	Decimal Adjust Accumulator	1	12
<b>LOGICAL OPERATIONS</b>				
ANL	A,Rn	AND Register to Accumulator	1	12
ANL	A,direct	AND direct byte to Accumulator	2	12
ANL	A,@Ri	AND indirect RAM to Accumulator	1	12
ANL	A,#data	AND immediate data to Accumulator	2	12
ANL	direct,A	AND Accumulator to direct byte	2	12
ANL	direct,#data	AND immediate data to direct byte	3	24
ORL	A,Rn	OR register to Accumulator	1	12
ORL	A,direct	OR direct byte to Accumulator	2	12
ORL	A,@Ri	OR indirect RAM to Accumulator	1	12
ORL	A,#data	OR immediate data to Accumulator	2	12
ORL	direct,A	OR Accumulator to direct byte	2	12
ORL	direct,#data	OR immediate data to direct byte	3	24
XRL	A,Rn	Exclusive-OR register to Accumulator	1	12
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	12
XRL	A,@Ri	Exclusive-OR indirect RAM to Accumulator	1	12
XRL	A,#data	Exclusive-OR immediate data to Accumulator	2	12
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	12
XRL	direct,#data	Exclusive-OR immediate data to direct byte	3	24
CLR	A	Clear Accumulator	1	12
CPL	A	Complement Accumulator	1	12
RL	A	Rotate Accumulator left	1	12
RLC	A	Rotate Accumulator left through the carry	1	12
RR	A	Rotate Accumulator right	1	12
RRC	A	Rotate Accumulator right through the carry	1	12
SWAP	A	Swap nibbles within the Accumulator	1	12
<b>DATA TRANSFER</b>				
MOV	A,Rn	Move register to Accumulator	1	12
MOV	A,direct	Move direct byte to Accumulator	2	12
MOV	A,@Ri	Move indirect RAM to Accumulator	1	12

All mnemonics copyrighted © Intel Corporation 1980

## 80C51 Family

80C51 family programmer's guide  
and instruction set

Table 7. 80C51 Instruction Set Summary (Continued)

MNEMONIC	BYTE	DESCRIPTION	BYTE	OSCILLATOR PERIOD
<b>DATA TRANSFER (Continued)</b>				
MOV A,#data	2	Move immediate data to Accumulator	2	12
MOV Rn,A	1	Move Accumulator to register	1	12
MOV Rn,direct	2	Move direct byte to register	2	12
MOV RN,#data	2	Move immediate data to register	2	12
MOV direct,A	2	Move Accumulator to direct byte	2	12
MOV direct,Rn	2	Move register to direct byte	2	24
MOV direct,direct	3	Move direct byte to direct	3	24
MOV direct,@Ri	2	Move indirect RAM to direct byte	2	24
MOV direct,#data	3	Move immediate data to direct byte	3	24
MOV @Ri,A	1	Move Accumulator to indirect RAM	1	12
MOV @Ri,direct	2	Move direct byte to indirect RAM	2	24
MOV @Ri,#data	2	Move immediate data to indirect RAM	2	12
MOV DPTR,#data16	3	Load Data Pointer with a 16-bit constant	3	24
MOVC A,A+DPTR	1	Move Code byte relative to DPTR to Acc	1	24
MOVC A,A+PC	1	Move Code byte relative to PC to Acc	1	24
MOVX A,@Ri	1	Move external RAM (8-bit addr) to Acc	1	24
MOVX A,DPTR	1	Move external RAM (16-bit addr) to Acc	1	24
MOVX A,@Ri,A	1	Move Acc to external RAM (8-bit addr)	1	24
MOVX @DPTR,A	1	Move Acc to external RAM (16-bit addr)	1	24
PUSH direct	2	Push direct byte onto stack	2	24
POP direct	2	Pop direct byte from stack	2	24
XCH A,Rn	1	Exchange register with Accumulator	1	12
XCH A,direct	2	Exchange direct byte with Accumulator	2	12
XCH A,@Ri	1	Exchange indirect RAM with Accumulator	1	12
XCHD A,@Ri	1	Exchange low-order digit indirect RAM with Acc	1	12
<b>BOOLEAN VARIABLE MANIPULATION</b>				
CLR C		Clear carry	1	12
CLR bit		Clear direct bit	2	12
SETB C		Set carry	1	12
SETB bit		Set direct bit	2	12
CPL C		Complement carry	1	12
CPL bit		Complement direct bit	2	12
ANL C,bit		AND direct bit to carry	2	24
ANL C,/bit		AND complement of direct bit to carry	2	24
ORL C,bit		OR direct bit to carry	2	24
ORL C,/bit		OR complement of direct bit to carry	2	24
MOV C,bit		Move direct bit to carry	2	12
MOV bit,C		Move carry to direct bit	2	24
JC rel		Jump if carry is set	2	24
JNC rel		Jump if carry not set	2	24

All mnemonics copyrighted © Intel Corporation 1980

## 80C51 Family

80C51 family programmer's guide  
and instruction set

Table 7. 80C51 Instruction Set Summary (Continued)

OSCILLATOR PERIOD	BYTE	DESCRIPTION	BYTE	MNEMONIC
<b>BOOLEAN VARIABLE MANIPULATION (Continued)</b>				
24	2	Jump if direct bit is set	2	JB rel
24	2	Jump if direct bit is not set	2	JNB rel
24	3	Jump if direct bit is set and clear bit	3	JBC bit,rel
<b>PROGRAM BRANCHING</b>				
24	2	Absolute subroutine call	2	ACALL addr11
24	3	Long subroutine call	3	LCALL addr16
24	1	Return from subroutine	1	RET
24	1	Return from interrupt	1	RETI
24	2	Absolute jump	2	AJMP addr11
24	3	Long jump	3	LJMP addr16
24	2	Short jump (relative addr)	2	SJMP rel
24	1	Jump indirect relative to the DPTR	1	JMP @A+DPTR
24	2	Jump if Accumulator is zero	2	JZ rel
24	2	Jump if Accumulator is not zero	2	JNZ rel
24	3	Compare direct byte to Acc and jump if not equal	3	CJNE A,direct,rel
24	3	Compare immediate to Acc and jump if not equal	3	CJNE A,#data,rel
24	3	Compare immediate to register and jump if not equal	3	CJNE Rn,#data,rel
24	3	Compare immediate to indirect and jump if not equal	3	CJNE @Ri,#data,rel
24	2	Decrement register and jump if not zero	2	DJNZ Rn,rel
24	3	Decrement direct byte and jump if not zero	3	DJNZ direct,rel
12	1	No operation	1	NOP

All mnemonics copyrighted © Intel Corporation 1980

## 80C51 Family

80C51 family programmer's guide  
and instruction set

## INSTRUCTION DEFINITIONS

**ACALL addr11****Function:** Absolute Call

**Description:** ACALL unconditionally calls a subroutine located at the indicated address. The instruction increments the PC twice to obtain the address of the following instruction, then pushes the 16-bit result onto the stack (low-order byte first) and increments the Stack Pointer twice. The destination address is obtained by successively concatenating the five high-order bits of the incremented PC, opcode bits 7-5, and the second byte of the instruction. The subroutine called must therefore start within the same 2k block of the program memory as the first byte of the instruction following ACALL. No flags are affected.

**Example:** Initially SP equals 07H. The label "SUBRTN" is at program memory location 0345 H. After executing the instruction,

ACALL SUBRTN

at location 0123H, SP will contain 09H, internal RAM locations 08H and 09H will contain 25H and 01H, respectively, and the PC will contain 0345H.

**Bytes:** 2**Cycles:** 2**Encoding:**

a10a9 a8 1

0 0 0 1

a7 a6 a5 a4

a3 a2 a1 a0

**Operation:**

ACALL

 $(PC) \leftarrow (PC) + 2$  $(SP) \leftarrow (SP) + 1$  $(SP) \leftarrow (PC_{7-0})$  $(SP) \leftarrow (SP) + 1$  $(SP) \leftarrow (PC_{15-8})$  $(PC_{10-0}) \leftarrow \text{page address}$

80C51 Family

80C51 family programmer's guide  
and instruction set

ADD A,<src-byte>

Function: Add

**Description:** ADD adds the byte variable indicated to the Accumulator, leaving the result in the Accumulator. The carry and auxiliary-carry flags are set, respectively, if there is a carry-out from bit 7 or bit 3, and cleared otherwise. When adding unsigned integers, the carry flag indicates an overflow occurred. OV is set if there is a carry-out of bit 6 but not out of bit 7, or a carry-out of bit 7 but not bit 6; otherwise OV is cleared. When adding signed integers, OV indicates a negative number produced as the sum of two positive operands, or a positive sum from two negative operands. Four source operand addressing modes are allowed: register, direct, register-indirect, or immediate.

**Example:** The Accumulator holds 0C3H (11000011B) and register 0 holds 0AAH (10101010B). The instruction, ADD A,R0 will leave 6DH (01101101B) in the Accumulator with the AC flag cleared and both the Carry flag and OV set to 1.

ADD A,Rn

Bytes: 1

Cycles: 1

Encoding: 0 0 1 0 1 r r r

Operation: ADD  
(A) ← (A) + (R<sub>n</sub>)

ADD A,direct

Bytes: 2

Cycles: 1

Encoding: 0 0 1 0 0 1 0 1 direct address

Operation: ADD  
(A) ← (A) + (direct)

ADD A,@Ri

Bytes: 1

Cycles: 1

Encoding: 0 0 1 0 0 1 1 i

Operation: ADD  
(A) ← (A) + ((R<sub>i</sub>))

ADD A,#data

Bytes: 2

Cycles: 1

Encoding: 0 0 1 0 0 1 0 0 immediate data

Operation: ADD  
(A) ← (A) + #data

## 80C51 Family

80C51 family programmer's guide  
and instruction set**ADDC A,<src-byte>****Function:** Add with Carry

**Description:** ADDC simultaneously adds the byte variable indicated, the carry flag and the Accumulator contents, leaving the result in the Accumulator. The carry and auxiliary-carry flags are set, respectively, if there is a carry-out from bit 7 or bit 3, and cleared otherwise. When adding unsigned integers, the carry flag indicates an overflow occurred.

OV is set if there is a carry-out of bit 6 but not out of bit 7, or a carry-out of bit 7 but not out of bit 6; otherwise OV is cleared. When adding signed integers, OV indicates a negative number produced as the sum of two positive operands, or a positive sum from two negative operands.

Four source operand addressing modes are allowed: register, direct, register-indirect, or immediate.

**Example:** The Accumulator holds 0C3H (11000011B) and register 0 holds 0AAH (10101010B) with the carry flag set. The instruction,

```
ADDC A,R0
```

will leave 6EH (01101110B) in the Accumulator with AC cleared and both the Carry flag and OV set to 1.

**ADDC A,Rn****Bytes:** 1**Cycles:** 1**Encoding:**

0	0	1	1	1	r	r	r
---	---	---	---	---	---	---	---

**Operation:** ADDC  
 $(A) \leftarrow (A) + (C) + (R_n)$

**ADDC A,direct****Bytes:** 2**Cycles:** 1**Encoding:**

0	0	1	1	0	1	0	1
---	---	---	---	---	---	---	---

direct address

**Operation:** ADDC  
 $(A) \leftarrow (A) + (C) + (\text{direct})$

**ADDC A,@Ri****Bytes:** 1**Cycles:** 1**Encoding:**

0	0	1	1	0	1	1	i
---	---	---	---	---	---	---	---

**Operation:** ADDC  
 $(A) \leftarrow (A) + (C) + ((R_i))$

**ADDC A,#data****Bytes:** 2**Cycles:** 1**Encoding:**

0	0	1	1	0	1	0	0
---	---	---	---	---	---	---	---

immediate data

**Operation:** ADDC  
 $(A) \leftarrow (A) + (C) + \#data$



80C51 Family

80C51 family programmer's guide  
and instruction set

AJMP addr11

Function: Absolute Jump

Description: AJMP transfers program execution to the indicated address, which is formed at run-time by concatenating the high-order five bits of the PC (after incrementing the PC twice), opcode bits 7-5, and the second byte of the instruction. The destination must therefore be within the same 2k block of program memory as the first byte of the instruction following AJMP.

Example: The label "JMPADR" is at program memory location 0123H. The instruction, AJMP JMPADR is at location 0345H and will load the PC with 0123H.

Bytes: 2

Cycles: 2

Encoding: 

a10	a9	a8	0	0	0	1	a7	a6	a5	a4	a3	a2	a1	a0
-----	----	----	---	---	---	---	----	----	----	----	----	----	----	----

Operation: AJMP  
(PC) ← (PC) + 2  
(PC<sub>10:0</sub>) ← page address

ANL <dest-byte>,<src-byte>

Function: Logical-AND for byte variables

Description: ANL performs the bitwise logical-AND operation between the variables indicated and stores the results in the destination variable. No flags are affected.  
  
The two operands allow six addressing mode combinations. When the destination is the Accumulator, the source can use register, direct, register-indirect, or immediate addressing; when the destination is a direct address, the source can be the Accumulator or immediate data.  
  
*Note:* When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.

Example: If the Accumulator holds 0C3H (11000011B) and register 0 holds 55H (01010101B) then the instruction, ANL A,R0 will leave 41H (0100001B) in the Accumulator.  
  
When the destination is a directly addressed byte, this instruction will clear combinations of bits in any RAM location or hardware register. The mask byte determining the pattern of bits to be cleared would either be a constant contained in the instruction or a value computed in the Accumulator at run-time. The instruction, ANL P1,#0111001B will clear bits 7, 3, and 2 of output port 1.

## 80C51 Family

80C51 family programmer's guide  
and instruction set

<b>ANL A,Rn</b>					
<b>Bytes:</b>	1				
<b>Cycles:</b>	1				
<b>Encoding:</b>	0 1 0 1	1 r r r r			
<b>Operation:</b>	ANL (A) ← (A) ∧ (R <sub>n</sub> )				
<b>ANL A,direct</b>					
<b>Bytes:</b>	2				
<b>Cycles:</b>	1				
<b>Encoding:</b>	0 1 0 1	0 1 0 1	direct address		
<b>Operation:</b>	ANL (A) ← (A) ∧ (direct)				
<b>ANL A,@Ri</b>					
<b>Bytes:</b>	1				
<b>Cycles:</b>	1				
<b>Encoding:</b>	0 1 0 1	0 1 1 i			
<b>Operation:</b>	ANL (A) ← (A) ∧ ((R <sub>i</sub> ))				
<b>ANL A,#data</b>					
<b>Bytes:</b>	2				
<b>Cycles:</b>	1				
<b>Encoding:</b>	0 1 0 1	0 1 0 0	immediate data		
<b>Operation:</b>	ANL (A) ← (A) ∧ #data				
<b>ANL direct,A</b>					
<b>Bytes:</b>	2				
<b>Cycles:</b>	1				
<b>Encoding:</b>	0 1 0 1	0 0 1 0	direct address		
<b>Operation:</b>	ANL (A) ← (direct) ∧ (A)				
<b>ANL direct,#data</b>					
<b>Bytes:</b>	3				
<b>Cycles:</b>	2				
<b>Encoding:</b>	0 1 0 1	0 0 1 1	direct address	immediate data	
<b>Operation:</b>	ANL (direct) ← (direct) ∧ #data				

80C51 Family

80C51 family programmer's guide  
and instruction set

ANL C,<src-bit>		ANL, A, Rn
Function:	Logical-AND for bit variables	Bytes: 1
Description:	If the Boolean value of the source bit is a logical 0 then clear the carry flag; otherwise leave the carry flag in its current state. A slash ("/") preceding the operand in the assembly language indicates that the logical complement of the addressed bit is used as the source value, <i>but the source bit itself is not affected</i> . No other flags are affected.	Operation: ANL A, direct
Example:	Only direct addressing is allowed for the source operand. Set the carry flag if, and only if, P1.0 = 1, ACC.7 = 1, and OV = 0: MOV C,P1.0 ;LOAD CARRY WITH INPUT PIN STATE ANL C,ACC.7;AND CARRY WITH ACCUM. BIT 7 ANL C,/OV ;AND WITH INVERSE OF OVERFLOW FLAG	Bytes: 2 Cycles: 1
ANL C,bit	<div>direct address</div> <div>0 1 0 1 0 1 0 1</div>	Encoding: ANL A, direct
Bytes:	2	Bytes: 2
Cycles:	2	Cycles: 1
Encoding:	<div>1 0 0 0 0 0 1 0</div> <div>bit address</div>	Operation: ANL A, direct
Operation:	ANL $(C) \leftarrow (C) \wedge (\text{bit})$	Bytes: 1 Cycles: 1
ANL C,bit	<div>0 1 0 1 0 1 0 1</div>	Encoding: ANL A, direct
Bytes:	2	Bytes: 2
Cycles:	2	Cycles: 1
Encoding:	<div>1 0 1 1 0 0 0 0</div> <div>bit address</div>	Operation: ANL A, direct
Operation:	ANL $(C) \leftarrow (C) \wedge \neg (\text{bit})$	Bytes: 1 Cycles: 1
	<div>immediate data</div> <div>0 1 0 1 0 1 0 0</div>	Encoding: ANL A, direct
		Bytes: 2 Cycles: 1
	<div>direct address</div> <div>0 1 0 1 0 0 1 0</div>	Encoding: ANL A, direct
		Bytes: 2 Cycles: 1
	<div>immediate data</div> <div>0 1 0 1 0 0 1 1</div>	Encoding: ANL A, direct
		Bytes: 2 Cycles: 1

80C51 Family

80C51 family programmer's guide  
and instruction set

CJNE <dest-byte>,<src-byte>,rel

**Function:** Compare and Jump if Not Equal

**Description:** CJNE compares the magnitudes of the first two operands, and branches if their values are not equal. The branch destination is computed by adding the signed relative-displacement in the last instruction byte to the PC, after incrementing the PC to the start of the next instruction. The carry flag is set if the unsigned integer value of <dest-byte> is less than the unsigned integer value of <src-byte>; otherwise, the carry is cleared. Neither operand is affected.

The first two operands allow four addressing mode combinations: the Accumulator may be compared with any directly addressed byte or immediate data, and any indirect RAM location or working register can be compared with an immediate constant.

**Example:** The Accumulator contains 34H. Register 7 contains 56H. The first instruction in the sequence,

```
CJNE    R7,#60H,NOT_EQ
;      ...
NOT_EQ  JC    REQ_LOW
;      ...
```

R7 = 60H.  
IF R7 < 60H.  
R7 > 60H.

sets the carry flag and branches to the instruction at label NOT\_EQ. By testing the carry flag, this instruction determines whether R7 is greater or less than 60H.

If the data being presented to Port 1 is also 34H, then the instruction,

WAIT: CJNE A,P1,WAIT

clears the carry flag and continues with the next instruction in sequence, since the Accumulator does equal the data read from P1. (If some other value was being input on P1, the program will loop at this point until the P1 data changes to 34H.)

CJNE A,direct,rel

**Bytes:** 3

**Cycles:** 2

**Encoding:**



**Operation:**

$(PC) \leftarrow (PC) + 3$   
IF (A) < > (direct)  
THEN  
 $(PC) \leftarrow (PC) + \text{relative offset}$   
IF (A) < (direct)  
THEN.  
ELSE  
 $(C) \leftarrow 1$   
 $(C) \leftarrow 0$

80C51 Family

80C51 family programmer's guide  
and instruction set

<b>CJNE A,#data,rel</b>	<b>CJNE &lt;dest-byte&gt;,&lt;src-byte&gt;,rel</b>
<b>Bytes:</b> 3	<b>Function:</b> Compare and Jump if Not Equal
<b>Cycles:</b> 2	<b>Description:</b> CJNE compares the magnitudes of the first two operands, and branches if their values are not equal. The first operand is compared by adding the signed relative displacement in the last instruction byte to the PC, after which the PC is incremented by one. The second operand is compared with the integer value of the first operand. Neither operand is affected.
<b>Encoding:</b> 1 0 1 1 0 1 0 0 immediate data rel. address	
<b>Operation:</b> (PC) ← (PC) + 3 IF (A) <> data THEN (PC) ← (PC) + relative offset	
<b>Example:</b> The Accumulator contains 34H. Register 7 contains 56H. The first instruction is CJNE R7,#80H,NOT_EQ. The carry flag and brancher to the instruction at label NOT_EQ. By testing the carry flag, the instruction determines whether R7 is greater or less than 80H. If the data being presented to Port 1 is also 34H, then the instruction WAIT: CJNE A,P1,WAIT.	
<b>CJNE Rn,#data,rel</b>	
<b>Bytes:</b> 3	
<b>Cycles:</b> 2	
<b>Encoding:</b> 1 0 1 1 1 r r r immediate data rel. address	
<b>Operation:</b> (PC) ← (PC) + 3 IF (R <sub>n</sub> ) <> data THEN (PC) ← (PC) + relative offset IF (R <sub>n</sub> ) < data THEN (C) ← 1 ELSE (C) ← 0	
<b>CJNE @Ri,#data,rel</b>	
<b>Bytes:</b> 3	
<b>Cycles:</b> 2	
<b>Encoding:</b> 1 0 1 1 0 1 1 i immediate data rel. address	
<b>Operation:</b> (PC) ← (PC) + 3 IF ((R <sub>i</sub> )) <> data THEN (PC) ← (PC) + relative offset IF ((R <sub>i</sub> )) < data THEN (C) ← 1 ELSE (C) ← 0	

## 80C51 Family

**CLR A****Function:** Clear Accumulator**Description:** The Accumulator is cleared (all bits reset to zero). No flags are affected.**Example:** The Accumulator contains 5CH (01011100B). The instruction,  
CLR A  
will leave the Accumulator set to 00H (00000000B).**Bytes:** 1**Cycles:** 1**Encoding:**

1	1	1	0	0	1	0	0
---	---	---	---	---	---	---	---

**Operation:**CLR A  
(A) ← 0**CLR bit****Function:** Clear bit**Description:** The indicated bit is cleared (reset to zero). No other flags are affected. CLR can operate on the carry flag or any directly addressable bit.**Example:** Port 1 has previously been written with 5DH (01011101B). The instruction,  
CLR P1.2  
will leave the port set to 59H (01011001B).**CLR C****Bytes:** 1**Cycles:** 1**Encoding:**

1	1	0	0	0	0	1	1
---	---	---	---	---	---	---	---

**Operation:**CLR C  
(C) ← 0**CLR bit****Bytes:** 2**Cycles:** 1**Encoding:**

1	1	0	0	0	0	1	0
---	---	---	---	---	---	---	---

bit address

**Operation:**CLR (bit)  
(bit) ← 0



## CPL A

**Function:** Complement Accumulator

**Description:** Each bit of the Accumulator is logically complemented (one's complement). Bits which previously contained a one are changed to a zero and vice-versa. No flags are affected.

**Example:** The Accumulator contains 5CH (01011100B). The instruction, CPL A will leave the Accumulator set to 0A3H (10100011B).

**Bytes:** 1

**Cycles:** 1

**Encoding:** 1 1 1 1 0 1 0 0

**Operation:** CPL  
(A)  $\leftarrow \neg$  (A)

## CPL bit

**Function:** Complement bit

**Description:** The bit variable specified is complemented. A bit which had been a one is changed to zero and vice-versa. No other flags are affected. CLR can operate on the carry or any directly addressable bit.  
*Note:* When this instruction is used to modify an output pin, the value used as the original data will be read from the output data latch, *not* the input pin.

**Example:** Port 1 has previously been written with 5DH (01011101B). The instruction sequence, CPL P1.1  
CPL P1.2  
will leave the port set to 5BH (01011011B).

## CPL C

**Bytes:** 1

**Cycles:** 1

**Encoding:** 1 0 1 1 0 0 1 1

**Operation:** CPL  
(C)  $\leftarrow \neg$  (C)

## CPL bit

**Bytes:** 2

**Cycles:** 1

**Encoding:** 1 0 1 1 0 0 1 0 bit address

**Operation:** CPL  
(bit)  $\leftarrow \neg$  (bit)

**DA A****Function:** Decimal-adjust Accumulator for Addition**Description:** DA A adjusts the eight-bit value in the Accumulator resulting from the earlier addition of two variables (each in packed-BCD format), producing two four-bit digits. Any ADD or ADDC instruction may have been used to perform the addition.

If Accumulator bits 3-0 are greater than nine (xxx1010-xxx1111), or if the AC flag is one, six is added to the Accumulator, producing the proper BCD digit in the low-order nibble. This internal addition would set the carry flag if a carry-out of the low-order four-bit field propagated through all high-order bits, but it would not clear the carry flag otherwise.

If the carry flag is now set, or if the four high-order bits now exceed nine (1010xxx-111xxxx), these high-order bits are incremented by six, producing the proper BCD digit in the high-order nibble. Again, this would set the carry flag if there was a carry-out of the high-order bits, but wouldn't clear the carry. The carry flag thus indicates if the sum of the original two BCD variables is greater than 100, allowing multiple precision decimal addition. OV is not affected.

All of this occurs during the one instruction cycle. Essentially, this instruction performs the decimal conversion by adding 00H, 06H, 60H, or 66H to the Accumulator, depending on initial Accumulator and PSW conditions.

*Note:* DA A *cannot* simply convert a hexadecimal number in the Accumulator to BCD notation, nor does DA A apply to decimal subtraction.

**Example:** The Accumulator holds the value 56H (01010110B) representing the packed BCD digits of the decimal number 56. Register 3 contains the value 67H (01100111B) representing the packed BCD digits of the decimal number 67. The carry flag is set. The instruction sequence,

```
ADDC    A,R3
DA      A
```

will first perform a standard two's-complement binary addition, resulting in the value 0BEH (10111110B) in the Accumulator. The carry and auxiliary carry flags will be cleared.

The Decimal Adjust instruction will then alter the Accumulator to the value 24H (00100100B), indicating the packed BCD digits of the decimal number 24, the low-order two digits of the decimal sum of 56, 67, and the carry-in. The carry flag will be set by the Decimal Adjust instruction, indicating that a decimal overflow occurred. The true sum 56, 67, and 1 is 124.

BCD variables can be incremented or decremented by adding 01H or 99H. If the Accumulator initially holds 30H (representing the digits of 30 decimal), the the instruction sequence,

```
ADD     A,#99H
DA      A
```

will leave the carry set and 29H in the Accumulator, since  $30 + 99 = 129$ . The low-order byte of the sum can be interpreted to mean  $30 - 1 = 29$ .

**Bytes:** 1**Cycles:** 1**Encoding:**

1	1	0	1	0	1	0	0
---	---	---	---	---	---	---	---

**Operation:**

```
DA
--contents of Accumulator are BCD
IF  [((A3-0) > 9) ∨ ((AC) = 1)]
    THEN(A3-0) ← (A3-0) + 6
    AND
IF  [((A7-4) > 9) ∨ ((C) = 1)]
    THEN(A7-4) ← (A7-4) + 6
```

80C51 family programmer's guide  
and instruction set

80C51 Family

DEC byte													
Function:	Decrement												
Description:	The variable indicated is decremented by 1. An original value of 00H will underflow to 0FFH. No flags are affected. Four operand addressing modes are allowed: accumulator, register, direct, or register-indirect.  <i>Note:</i> When this instruction is used to modify an output port, the value used as the original data will be read from the output data latch, <i>not</i> the input pin.												
Example:	Register 0 contains 7FH (01111111B). Internal RAM locations 7EH and 7FH contain 00H and 40H, respectively. The instruction sequence, <div><div>DEC @R0</div><div>DEC R0</div><div>DEC @R0</div></div> will leave register 0 set to 7EH and internal RAM locations 7EH and 7FH set to 0FFH and 3FH.												
DEC A													
Bytes:	1												
Cycles:	1												
Encoding:	<table><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr></table>	0	0	0	1								
0	0	0	1										
Operation:	DEC (A) ← (A) - 1												
DEC Rn													
Bytes:	1												
Cycles:	1												
Encoding:	<table><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td></td><td>1</td><td>r</td><td>r</td></tr></table>	0	0	0	1		1	r	r				
0	0	0	1										
	1	r	r										
Operation:	DEC (R <sub>n</sub> ) ← (R <sub>n</sub> ) - 1												
DEC direct													
Bytes:	2												
Cycles:	1												
Encoding:	<table><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td></td><td>0</td><td>1</td><td>0</td></tr></table> <table><tr><td>direct address</td></tr></table>	0	0	0	1		0	1	0	direct address			
0	0	0	1										
	0	1	0										
direct address													
Operation:	DEC (direct) ← (direct) - 1												
DEC @Ri													
Bytes:	1												
Cycles:	1												
Encoding:	<table><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td></td><td>0</td><td>1</td><td>1</td></tr><tr><td></td><td></td><td></td><td>i</td></tr></table>	0	0	0	1		0	1	1				i
0	0	0	1										
	0	1	1										
			i										
Operation:	DEC ((R <sub>i</sub> )) ← ((R <sub>i</sub> )) - 1												

## 80C51 Family

80C51 family programmer's guide  
and instruction set**DIV AB****Function:** Divide**Description:** DIV AB divides the unsigned eight-bit integer in the Accumulator by the unsigned eight-bit integer in register B.

The Accumulator receives the integer part of the quotient; register B receives the integer remainder. The carry and OV flags will be cleared.

**Exception:** if B had originally contained 00H, the values returned in the Accumulator and B-register will be undefined and the overflow flag will be set. The carry flag is cleared in any case.**Example:** The Accumulator contains 251 (0FBH or 11111011B) and B contains 18 (12H or 00010010B). The instruction,

DIV AB

will leave 13 in the Accumulator (0DH or 00001101B) and the value 17 (11H or 00010001B) in B, since  $251 = (13 \times 18) + 17$ . Carry and OV will both be cleared.**Bytes:** 1**Cycles:** 4**Encoding:**

1	0	0	0	0	1	0	0
---	---	---	---	---	---	---	---

**Operation:** DIV  
 $(A)_{15:8} \leftarrow (A)/(B)$   
 $(B)_{7:0}$

80C51 Family

80C51 family programmer's guide  
and instruction set

DJNZ <byte>,<rel-addr>

**Function:** Decrement and Jump if Not Zero

**Description:** DJNZ decrements the location indicated by 1, and branches to the address indicated by the second operand if the resulting value is not zero. An original value of 00H will underflow to 0FFH. No flags are affected. The branch destination would be computed by adding the signed relative-displacement value in the last instruction byte to the PC, after incrementing the PC to the first byte of the following instruction.

The location decremented may be a register or directly addressed byte.

*Note:* When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.

**Example:** Internal RAM locations 40H, 50H, and 60H contain the values 01H, 70H, and 15H, respectively. The instruction sequence,

```
DJNZ 40H,LABEL_1
DJNZ 50H,LABEL_2
DJNZ 60H,LABEL_3
```

will cause a jump to the instruction at LABEL\_2 with the values 00h, 6FH, and 15H in the three RAM locations. The first jump was *not* taken because the result was zero.

This instruction provides a simple way of executing a program loop a given number of times, or for adding a moderate time delay (from 2 to 512 machine cycles) with a single instruction. The instruction sequence,

```
MOV R2,#8
TOGGLE: CPL P1.7
        DJNZ R2,TOGGLE
```

will toggle P1.7 eight times, causing four output pulses to appear at bit 7 of output Port 1. Each pulse will last three machine cycles, two for DJNZ and one to alter the pin.

DJNZ Rn,rel

**Bytes:** 2

**Cycles:** 2

**Encoding:**

1	1	0	1	1	r	r	r
---	---	---	---	---	---	---	---

rel. address
--------------

**Operation:** DJNZ  
 $(PC) \leftarrow (PC) + 2$   
 $(R_n) \leftarrow (R_n) - 1$   
IF  $(R_n) > 0$  or  $(R_n) < 0$   
THEN  
 $(PC) \leftarrow (PC) + rel$

DJNZ direct,rel

**Bytes:** 3

**Cycles:** 2

**Encoding:**

1	1	0	1	0	1	0	1
---	---	---	---	---	---	---	---

direct data
-------------

rel. address
--------------

**Operation:** DJNZ  
 $(PC) \leftarrow (PC) + 2$   
 $(direct) \leftarrow (direct) - 1$   
IF  $(direct) > 0$  or  $(direct) < 0$   
THEN  
 $(PC) \leftarrow (PC) + rel$

## 80C51 Family

80C51 family programmer's guide  
and instruction set**INC <byte>****Function:** Increment**Description:** INC increments the indicated variable by 1. An original value of 0FFH will overflow to 00H. No flags are affected. Three addressing modes are allowed: register, direct, or register-indirect.**Note:** When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.**Example:** Register 0 contains 7EH (0111110B). Internal RAM locations 7EH and 7FH contain 0FFH and 40H, respectively. The instruction sequence,

```

INC    @R0
INC    R0
INC    @R0

```

will leave register 0 set to 7FH and internal RAM locations 7EH and 7FH holding (respectively) 00H and 41H.

**INC A****Bytes:** 1**Cycles:** 1**Encoding:**

0	0	0	0	0	1	0	0
---	---	---	---	---	---	---	---

**Operation:** INC $(A) \leftarrow (A) + 1$ **INC Rn****Bytes:** 1**Cycles:** 1**Encoding:**

0	0	0	0	1	r	r	r
---	---	---	---	---	---	---	---

**Operation:** INC $(R_n) \leftarrow (R_n) + 1$ **INC direct****Bytes:** 2**Cycles:** 1**Encoding:**

0	0	0	0	0	1	0	1	direct address
---	---	---	---	---	---	---	---	----------------

**Operation:** INC $(\text{direct}) \leftarrow (\text{direct}) + 1$ **INC @Ri****Bytes:** 1**Cycles:** 1**Encoding:**

0	0	0	0	0	1	1	i
---	---	---	---	---	---	---	---

**Operation:** INC $((R_i)) \leftarrow ((R_i)) + 1$



80C51 Family

80C51 family programmer's guide  
and instruction set

INC DPTR

Function: Increment Data Pointer

Description: Increment the 16-bit data pointer by 1. A 16-bit increment (modulo 2<sup>16</sup>) is performed; an overflow of the low-order byte of the data pointer (DPL) from 0FFH to 00H will increment the high-order byte (DPH). No flags are affected.  
This is the only 16-bit register which can be incremented.

Example: Registers DPH and DPL contain 12H and 0FEH, respectively. The instruction sequence,  
INC DPTR  
INC DPTR  
INC DPTR  
will change DPH and DPL to 13H and 01H.

Bytes: 1

Cycles: 2

Encoding: 1 0 1 0 0 0 1 1

Operation: INC  
(DPTR) ← (DPTR) + 1

JB bit,rel

Function: Jump if Bit set

Description: If the indicated bit is a one, jump to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. *The bit tested is not modified.* No flags are affected.

Example: The data present at input port 1 is 11001010B. The Accumulator holds 56 (01010110B). The instruction sequence,  
JB P1.2,LABEL1  
JB ACC.2,LABEL2  
will cause program execution to branch to the instruction at label LABEL2.

Bytes: 3

Cycles: 2

Encoding: 0 0 1 0 0 0 0 0 bit address rel. address

Operation: JB  
(PC) ← (PC) + 3  
IF (bit) = 1  
THEN  
(PC) ← (PC) + rel

## 80C51 Family

80C51 family programmer's guide  
and instruction set**JBC bit,rel****Function:** Jump if Bit is set and Clear bit**Description:** If the indicated bit is a one, branch to the address indicated; otherwise proceed with the next instruction. *The bit will not be cleared if it is already a zero.* The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. No flags are affected.

Note: When this instruction is used to test an output pin, the value used as the original data will read from the output data latch, not the input pin.

**Example:** The Accumulator holds 56H (01010110B). The instruction sequence;

```

JBC  ACC.3,LABEL1
JBC  ACC.2,LABEL2

```

will cause program execution to continue at the instruction identified by the LABEL2, with the Accumulator modified to 52H (01010010B).

**Bytes:** 3**Cycles:** 2**Encoding:**

0	0	0	1	0	0	0	0
---	---	---	---	---	---	---	---

bit address

rel. address

**Operation:**

```

JBC
(PC) ← (PC) + 3
IF (bit) = 1
  THEN
    (bit) ← 0
    (PC) ← (PC) + rel

```

**JC rel****Function:** Jump if Carry is set**Description:** If the carry flag is set, branch to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. No flags are affected.**Example:** The carry flag is cleared. The instruction sequence,

```

JC  LABEL1
CPL C
JC  LABEL2

```

will set the carry and cause program execution to continue at the instruction identified by the label LABEL2.

**Bytes:** 2**Cycles:** 2**Encoding:**

0	1	0	0	0	0	0	0
---	---	---	---	---	---	---	---

rel. address

**Operation:**

```

JC
(PC) ← (PC) + 2
IF (C) = 1
  THEN
    (PC) ← (PC) + rel

```

80C51 family programmer's guide

80C51 family programmer's guide  
and instruction set

JMP @A+DPTR

Function:

Jump indirect

Description:

Add the eight-bit unsigned contents of the Accumulator with the sixteen-bit data pointer, and load the resulting sum to the program counter. This will be the address for subsequent instruction fetches. Sixteen-bit addition is performed (modulo  $2^{16}$ ): a carry-out from the low-order eight bits propagates through the higher-order bits. Neither the Accumulator nor the Data Pointer is altered. No flags are affected.

Example:

An even number from 0 to 6 is in the Accumulator. The following sequence of instructions will branch to one of four AJMP instructions in a jump table starting at JMP\_TBL:

JMP\_TBL:

MOV DPTR,#JMP\_TBL  
JMP @A+DPTR  
AJMP LABEL0  
AJMP LABEL1  
AJMP LABEL2  
AJMP LABEL3

Bytes:

1

Cycles:

2

Encoding:

0 1 1 1 0 0 1 1

Operation:

JMP  
(PC) ← (A) + (DPTR)

If the Accumulator equals 04H when starting this sequence, execution will jump to label LABEL2. Remember that AJMP is a two-byte instruction, so the jump instructions start at every other address.

JNB bit,rel

Function:

Jump if Bit Not set

Description:

If the indicated bit is a zero, branch to the indicated address; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. *The bit tested is not modified.* No flags are affected.

Example:

The data present at input port 1 is 11001010B. The Accumulator holds 56H (01010110B). The instruction sequence,

JNB P1.3,LABEL1  
JNB ACC.3,LABEL2

will cause program execution to continue at the instruction at label LABEL2.

Bytes:

3

Cycles:

2

Encoding:

0 0 1 1 0 0 0 0 bit address rel. address

Operation:

JNB  
(PC) ← (PC) + 3  
IF (bit) = 0  
THEN  
(PC) ← (PC) + rel

## 80C51 Family

80C51 family programmer's guide  
and instruction set**JNC rel****Function:** Jump if Carry Not set**Description:** If the carry flag is a zero, branch to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice to point to the next instruction. The carry flag is not modified.**Example:** The carry flag is set. The instruction sequence,

```

JNC LABEL1
CPL C
JNC LABEL2

```

will clear the carry and cause program execution to continue at the instruction identified by the label LABEL2.

**Bytes:** 2**Cycles:** 2**Encoding:**

0	1	0	1	0	0	0	0
---	---	---	---	---	---	---	---

rel. address
--------------

**Operation:** JNC  
 $(PC) \leftarrow (PC) + 2$   
 IF  $(C) = 0$   
 THEN  
 $(PC) \leftarrow (PC) + \text{rel}$

**JNZ rel****Function:** Jump if Accumulator Not Zero**Description:** If any bit of the Accumulator is a one, branch to the indicated address; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. The Accumulator is not modified. No flags are affected.**Example:** The Accumulator originally holds 00H. The instruction sequence,

```

JNZ LABEL1
INC A
JNZ LABEL2

```

will set the Accumulator to 01H and continue at label LABEL2.

**Bytes:** 2**Cycles:** 2**Encoding:**

0	1	1	1	0	0	0	0
---	---	---	---	---	---	---	---

rel. address
--------------

**Operation:** JNZ  
 $(PC) \leftarrow (PC) + 2$   
 IF  $A \neq 0$   
 THEN  $(PC) \leftarrow (PC) + \text{rel}$

80C51 Family

80C51 family programmer's guide  
and instruction set

<b>JZ rel</b>													
<b>Function:</b>	Jump if Accumulator Zero												
<b>Description:</b>	If all bits of the Accumulator are zero, branch to the indicated address; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. The Accumulator is not modified. No flags are affected.												
<b>Example:</b>	The Accumulator originally holds 01H. The instruction sequence, JZ LABEL1 DEC A JZ LABEL2 will change the Accumulator to 00H and cause program execution to continue at the instruction identified by the label LABEL2.												
<b>Bytes:</b>	2												
<b>Cycles:</b>	2												
<b>Encoding:</b>	<table><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td colspan="4">rel. address</td></tr></table>	0	1	1	0	0	0	0	0	rel. address			
0	1	1	0										
0	0	0	0										
rel. address													
<b>Operation:</b>	JZ (PC) ← (PC) + 2 IF A = 0 THEN (PC) ← (PC) + rel												

<b>LCALL addr16</b>													
<b>Function:</b>	Long Call												
<b>Description:</b>	LCALL calls a subroutine located at the indicated address. The instruction adds three to the program counter to generate the address of the next instruction and then pushes the 16-bit result onto the stack (low byte first), incrementing the Stack Pointer by two. The high-order and low-order bytes of the PC are then loaded, respectively, with the second and third bytes of the LCALL instruction. Program execution continues with the instruction at this address. The subroutine may therefore begin anywhere in the full 64k-byte program memory address space. No flags are affected.												
<b>Example:</b>	Initially the Stack Pointer equals 07H. The label "SUBRTN" is assigned to program memory location 1234H. After executing the instruction, LCALL SUBRTN at location 0123H, the Stack Pointer will contain 09H, internal RAM locations 08H and 09H will contain 26H and 01H, and the PC will contain 1235H.												
<b>Bytes:</b>	3												
<b>Cycles:</b>	2												
<b>Encoding:</b>	<table><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td colspan="2">addr15-addr8</td><td colspan="2">addr7-addr0</td></tr></table>	0	0	0	1	0	0	1	0	addr15-addr8		addr7-addr0	
0	0	0	1										
0	0	1	0										
addr15-addr8		addr7-addr0											
<b>Operation:</b>	LCALL (PC) ← (PC) + 3 (SP) ← (SP) + 1 ((SP)) ← (PC <sub>7-0</sub> ) (SP) ← (SP) + 1 ((SP)) ← (PC <sub>15-8</sub> ) (PC) ← addr <sub>15-0</sub>												

## 80C51 Family

80C51 family programmer's guide  
and instruction set**LJMP addr16 (Implemented in 87C751 and 87C752 for in-circuit emulation only.)****Function:** Long Jump**Description:** LJMP causes an unconditional branch to the indicated address, by loading the high-order and low-order bytes of the PC (respectively) with the second and third instruction bytes. The destination may therefore be anywhere in the full 64k program memory address space. No flags are affected.**Example:** The label "JMPADR" is assigned to the instruction at program memory location 1234H. The instruction, LJMP JMPADR at location 0123H will load the program counter with 1234H.**Bytes:** 3**Cycles:** 2**Encoding:**

0 0 0 0 0 0 1 0

addr15-addr8

addr7-addr0

**Operation:**LJMP  
(PC) ← addr<sub>15-0</sub>**MOV <dest-byte>,<src-byte>****Function:** Move byte variable**Description:** The byte variable indicated by the second operand is copied into the location specified by the first operand. The source byte is not affected. No other register or flag is affected.

This is by far the most flexible operation. Fifteen combinations of source and destination addressing modes are allowed.

**Example:** Internal RAM location 30H holds 40H. The value of RAM location 40H is 10H. The data present at input port 1 is 11001010B (0CAH). The instruction sequence,

```

MOV R0,#30H ;R0 <= 30H
MOV A,@R0 ;A <= 40H
MOV R1,A ;R1 <= 40H
MOV B,@R1 ;B <= 10H
MOV @R1,P1 ;RAM (40H) <= 0CAH
MOV P2,P1 ;P2 #0CAH

```

leaves the value 30H in register 0, 40H in both the Accumulator and register 1, 10H in register B, and 0CAH (11001010B) both in RAM location 40H and output on port 2.

**MOV A,Rn****Bytes:** 1**Cycles:** 1**Encoding:**

1 1 1 0 1 r r r

**Operation:**MOV  
(A) ← (R<sub>n</sub>)



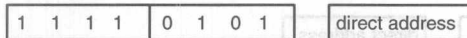
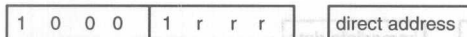
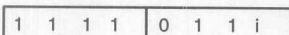
80C51 Family

80C51 family programmer's guide  
and instruction set

<b>*MOV A,direct</b>		
Bytes:	2	
Cycles:	1	
Encoding:	<div>11100101</div>	direct address
Operation:	MOV (A) ← (direct)	
<b>MOV A,@Ri</b>		
Bytes:	1	
Cycles:	1	
Encoding:	<div>1110011i</div>	
Operation:	MOV (A) ← ((R <sub>i</sub> ))	
<b>MOV A,#data</b>		
Bytes:	2	
Cycles:	1	
Encoding:	<div>01110100</div>	immediate data
Operation:	MOV (A) ← #data	
<b>MOV Rn,A</b>		
Bytes:	1	
Cycles:	1	
Encoding:	<div>11111rrr</div>	
Operation:	MOV (R <sub>n</sub> ) ← (A)	
<b>MOV Rn,direct</b>		
Bytes:	2	
Cycles:	2	
Encoding:	<div>10101rrr</div>	direct address
Operation:	MOV (R <sub>n</sub> ) ← (direct)	
<b>MOV Rn,#data</b>		
Bytes:	2	
Cycles:	1	
Encoding:	<div>01111rrr</div>	immediate data
Operation:	MOV (R <sub>n</sub> ) ← #data	

\*MOV A,ACC is not a valid instruction.

## 80C51 Family

80C51 family programmer's guide  
and instruction set**MOV direct,A****Bytes:** 2**Cycles:** 1**Encoding:****Operation:**MOV  
(direct)  $\leftarrow$  (A)**MOV direct,Rn****Bytes:** 2**Cycles:** 2**Encoding:****Operation:**MOV  
(direct)  $\leftarrow$  (R<sub>n</sub>)**MOV direct,direct****Bytes:** 3**Cycles:** 2**Encoding:****Operation:**MOV  
(direct)  $\leftarrow$  (direct)**MOV direct,@Ri****Bytes:** 2**Cycles:** 2**Encoding:****Operation:**MOV  
(direct)  $\leftarrow$  ((R<sub>i</sub>))**MOV direct,#data****Bytes:** 3**Cycles:** 2**Encoding:****Operation:**MOV  
(direct)  $\leftarrow$  #data**MOV @Ri,A****Bytes:** 1**Cycles:** 1**Encoding:****Operation:**MOV  
((R<sub>i</sub>))  $\leftarrow$  (A)

80C51 Family

80C51 family programmer's guide  
and instruction set

MOV @Ri,direct

Bytes: 2

Cycles: 2

Encoding: 1 0 1 0 0 1 1 i direct address

Operation: MOV ((R<sub>i</sub>)) ← (direct)

MOV @Ri,#data

Bytes: 2

Cycles: 1

Encoding: 0 1 1 1 0 1 1 i immediate data

Operation: MOV ((R<sub>i</sub>)) ← #data

MOV <dest-bit>,<src-bit>

Function: Move bit data

Description: The Boolean variable indicated by the second operand is copied into the location specified by the first operand. One of the operands must be the carry flag; the other may be any directly addressable bit. No other register or flag is affected.

Example: The carry flag is originally set. The data present at input Port 3 is 11000101B. The data previously written to output Port 1 is 35H (00110101B). The instruction sequence,

MOV P1.3,C  
MOV C,P3.3  
MOV P1.2,C

will leave the carry cleared and change Port 1 to 39H (00111001B).

MOV C,bit

Bytes: 2

Cycles: 1

Encoding: 1 0 1 0 0 0 1 0 bit address

Operation: MOV (C) ← (bit)

MOV bit,C

Bytes: 2

Cycles: 2

Encoding: 1 0 0 1 0 0 1 0 bit address

Operation: MOV (bit) ← (C)

## 80C51 Family

80C51 family programmer's guide  
and instruction set**MOV DPTR,#data16****Function:** Load Data Pointer with a 16-bit constant**Description:** The Data Pointer is loaded with the 16-bit constant indicated. The 16-bit constant is loaded into the second and third bytes of the instruction. The second byte (DPH) is the high-order byte, while the third byte (DPL) holds the low-order byte. No flags are affected.

This is the only instruction which moves 16 bits of data at once.

**Example:** The instruction,

MOV DPTR,#1234H

will load the value 1234H into the Data Pointer: DPH will hold 12H and DPL will hold 34H.

**Bytes:** 3**Cycles:** 2**Encoding:**

1	0	0	1	0	0	0	0	immed. data15-8	immed. data7-0
---	---	---	---	---	---	---	---	-----------------	----------------

**Operation:**

MOV

(DPTR) ← (#data<sub>15-0</sub>)DPH ← #data<sub>15-8</sub> DPL ← #data<sub>7-0</sub>**MOVC A,@A+<base-reg>****Function:** Move Code byte**Description:** The MOVC instructions load the Accumulator with a code byte, or constant from program memory. The address of the byte fetched is the sum of the original unsigned eight-bit Accumulator contents and the contents of a sixteen-bit base register, which may be either the Data Pointer or the PC. In the latter case, the PC is incremented to the address of the following instruction before being added with the Accumulator; otherwise the base register is not altered. Sixteen-bit addition is performed so a carry-out from the low-order eight bits may propagate through higher-order bits. No flags are affected.**Example:** A value between 0 and 3 is in the Accumulator. The following instructions will translate the value in the Accumulator to one of four values defined by the DB (define byte) directive:

```
REL_PC:  INC  A
          MOVC A,@A+PC
          RET
          DB   66H
          DB   77H
          DB   88H
          DB   99H
```

If the subroutine is called with the Accumulator equal to 01H, it will return with 77H in the Accumulator. The INC A before the MOVC instruction is needed to "get around" the RET instruction above the table. If several bytes of code separated the MOVC from the table, the corresponding number would be added to the Accumulator instead.

**MOVC A,@A+DPTR****Bytes:** 1**Cycles:** 2**Encoding:**

1	0	0	1	0	0	1	1
---	---	---	---	---	---	---	---

**Operation:**

MOVC

(A) ← ((A) + (DPTR))

80C51 family programmer's guide  
and instruction set

80C51 family programmer's guide  
and instruction set

MOVX A,@A+PC		MOV DPH,DPTR
Bytes:	1	Function: Load Data Pointer with a 16-bit constant
Cycles:	2	Description: The Data Pointer is loaded with the 16-bit constant indicated. The 16-bit constant is loaded in the second and third bytes of the instruction. The second byte (DPH) is the high-order byte and the third byte (DPL) holds the low-order byte. No flags are affected by this instruction.
Encoding:	<div>100000011</div>	Example: MOV DPH,#1234H This instruction will load the value 1234H into the Data Pointer. DPH will hold 12H and DPL will hold 34H.
Operation:	MOVX (PC) ← (PC) + 1 (A) ← ((A) + (PC))	Bytes: 2 Cycles: 2
MOVX <dest-byte>,<src-byte> (Not implemented in the 8XC752 or 8XC752)		
Function:	Move External	Encoding: 10010000
Description:	<p>The MOVX instructions transfer data between the Accumulator and a byte of external data memory, hence the "X" appended to MOV. There are two types of instructions, differing in whether they provide an eight-bit or sixteen-bit indirect address to the external data RAM.</p> <p>In the first type, the contents of R0 or R1 in the current register bank provide an eight-bit address multiplexed with data on P0. Eight bits are sufficient for external I/O expansion decoding or for a relatively small RAM array. For somewhat larger arrays, port pins can be used to output higher-order address bits. These pins would be controlled by an output instruction preceding the MOVX.</p> <p>In the second type of MOVX instruction, The Data Pointer generates a sixteen-bit address. P2 outputs the high-order eight address bits (the contents of DPH) while P0 multiplexes the low-order eight bits (DPL) with data. The P2 Special Function Register retains its previous contents while the P2 output buffers are emitting the contents of DPH. This form is faster and more efficient when accessing very large data arrays (up to 64k bytes), since no additional instructions are needed to set up the output ports.</p> <p>It is possible in some situations to mix the two MOVX types. A large RAM array with its high-order address lines driven by P2 can be addressed via the Data Pointer, or with code to output high-order address bits to P2 followed by a MOVX instruction using R0 or R1.</p>	
Example:	An external 256 byte RAM using multiplexed address/data lines is connected to the 8051 Port 0. Port 3 provides control lines for the external RAM. Ports 1 and 2 are used for normal I/O. Registers 0 and 1 contain 12H and 34H. Location 34H of the external RAM holds the value 56H. The instruction sequence,  MOVX A,@R1 MOVX @R0,A  copies the value 56H into both the Accumulator and external RAM location 12H.	
MOVX A,@Ri		
Bytes:	1	INC A
Cycles:	2	MOVX A,@A+PC
Encoding:	<div>1110001i</div>	RET
Operation:	MOVX (A) ← ((Ri))	MOVX A,@DPTR
MOVX A,@DPTR		
Bytes:	1	Bytes: 1
Cycles:	2	Cycles: 2
Encoding:	<div>11100000</div>	Encoding: 10010001
Operation:	MOVX (A) ← ((DPTR))	Operation: MOVX (A) ← ((A) + (DPTR))

## 80C51 Family

**MOVX @Ri,A****Bytes:** 1**Cycles:** 2**Encoding:**

1	1	1	1	0	0	1	i
---	---	---	---	---	---	---	---

**Operation:** MOVX  
((Ri)) ← (A)**MOVX @DPTR,A****Bytes:** 1**Cycles:** 2**Encoding:**

1	1	1	1	0	0	0	0
---	---	---	---	---	---	---	---

**Operation:** MOVX  
((DPTR)) ← (A)**MUL AB****Function:** Multiply**Description:** MUL AB multiplies the unsigned eight-bit integers in the Accumulator and register B. The low-order byte of the sixteen-bit product is left in the Accumulator, and the high-order byte in B. If the product is greater than 255 (0FFH) the overflow flag is set; otherwise it is cleared. The carry flag is always cleared.**Example:** Originally the Accumulator holds the value 80 (50H). Register B holds the value 160 (0A0H). The instruction,

MUL AB

will give the product 12,800 (3200H), so B is changed to 32H (00110010B) and the Accumulator is cleared. The overflow flag is set, carry is cleared.

**Bytes:** 1**Cycles:** 4**Encoding:**

1	0	1	0	0	1	0	0
---	---	---	---	---	---	---	---

**Operation:** MUL  
(A)<sub>7:0</sub> ← (A) × (B)  
(B)<sub>15:8</sub>



## 80C51 Family

80C51 family programmer's guide  
and instruction set

## NOP

**Function:** No Operation**Description:** Execution continues at the following instruction. Other than the PC, no registers or flags are affected.**Example:** It is desired to produce a low-going output pulse on bit 7 of Port 2 lasting exactly 5 cycles. A simple SETB/CLR sequence would generate a one-cycle pulse, so four additional cycles must be inserted. This may be done (assuming are enabled) with the instruction sequence,

```

CLR    P2.7
NOP
NOP
NOP
NOP
SETB   P2.7

```

**Bytes:** 1**Cycles:** 1**Encoding:**

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

**Operation:**

```

NOP
(PC) ← (PC) + 1

```

## ORL &lt;dest-byte&gt;,&lt;src-byte&gt;

**Function:** Logical-OR for byte variables**Description:** ORL performs the bitwise logical-OR operation between the indicated variables, storing the results in the destination byte. No flags are affected.

The two operands allow six addressing mode combinations. When the destination is the Accumulator, the source can use register, direct, register-indirect, or immediate addressing; when the destination is a direct address, the source can be the Accumulator or immediate data.

**Note:** When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.

**Example:** If the Accumulator holds 0C3H (11000011B) and R0 holds 55H (01010101B) then the instruction,

```

ORL    A,R0

```

will leave the Accumulator holding the value 0D7H (11010111B).

When the destination is a directly addressed byte, the instruction can set combinations of bits in any RAM location or hardware register. The pattern of bits to be set is determined by a mask byte, which may be either a constant data value in the instruction or a variable computed in the Accumulator at run-time. The instruction,

```

ORL    P1,#00110010B

```

will set bits 5, 4, and 1 of output Port 1.

## ORL A,Rn

**Bytes:** 1**Cycles:** 1**Encoding:**

0	1	0	0	1	r	r	r
---	---	---	---	---	---	---	---

**Operation:**

```

ORL
(A) ← (A) ∨ (Rn)

```

## 80C51 Family

80C51 family programmer's guide  
and instruction set**ORL A,direct****Bytes:** 2**Cycles:** 1**Encoding:** 0 1 0 0 0 1 0 1 direct address**Operation:** ORL  
(A)  $\leftarrow$  (A)  $\vee$  (direct)**ORL A,@Ri****Bytes:** 1**Cycles:** 1**Encoding:** 0 1 0 0 0 1 1 i**Operation:** ORL  
(A)  $\leftarrow$  (A)  $\vee$  ((R<sub>i</sub>))**ORL A,#data****Bytes:** 2**Cycles:** 1**Encoding:** 0 1 0 0 0 1 0 0 immediate data**Operation:** ORL  
(A)  $\leftarrow$  (A)  $\vee$  #data**ORL direct,A****Bytes:** 2**Cycles:** 1**Encoding:** 0 1 0 0 0 0 1 0 direct address**Operation:** ORL  
(direct)  $\leftarrow$  (direct)  $\vee$  (A)**ORL direct,#data****Bytes:** 3**Cycles:** 2**Encoding:** 0 1 0 0 0 0 1 1 direct address immediate data**Operation:** ORL  
(direct)  $\leftarrow$  (direct)  $\vee$  #data

80C51 family programmer's guide  
and instruction set

5 of 9

ate otherwise. A slash  
plement of the  
No other flags are

## 80C51 Family

80C51 family programmer's guide  
and instruction set**POP direct****Function:** Pop from stack**Description:** The contents of the internal RAM location addressed by the Stack Pointer is read, and the Stack Pointer is decremented by one. The value read is then transferred to the directly addressed byte indicated. No flags are affected.**Example:** The Stack Pointer originally contains the value 32H, and internal RAM locations 30H through 32H contain the values 20H, 23H, and 01H, respectively. The instruction sequence,

POP DPH

POP DPL

will leave the Stack Pointer equal to the value 30H and the Data Pointer set to 0123H. At this point the instruction,

POP SP

will leave the Stack Pointer set to 20H. Note that in this special case the Stack Pointer was decremented to 2FH before being loaded with the value popped (20H).

**Bytes:** 2**Cycles:** 2**Encoding:**

1 1 0 1 0 0 0 0

direct address

**Operation:**

POP

 $(\text{direct}) \leftarrow ((\text{SP}))$  $(\text{SP}) \leftarrow (\text{SP}) - 1$ **PUSH direct****Function:** Push onto stack**Description:** The Stack Pointer is incremented by one. The contents of the indicated variable is then copied into the internal RAM location addressed by the Stack Pointer. Otherwise no flags are affected.**Example:** On entering an interrupt routine the Stack Pointer contains 09H. The Data Pointer holds the value 0123H. The instruction sequence,

PUSH DPL

PUSH DPH

will leave the Stack Pointer set to 0BH and store 23H and 01H in internal RAM locations 0AH and 0BH, respectively.

**Bytes:** 2**Cycles:** 2**Encoding:**

1 1 0 0 0 0 0 0

direct address

**Operation:**

PUSH

 $(\text{SP}) \leftarrow (\text{SP}) + 1$  $((\text{SP})) \leftarrow (\text{direct})$

## 80C51 Family

80C51 family programmer's guide  
and instruction set**RET****Function:** Return from subroutine**Description:** RET pops the high- and low-order bytes of the PC successively from the stack, decrementing the Stack Pointer by two. Program execution continues at the resulting address, generally the instruction immediately following an ACALL or LCALL. No flags are affected.**Example:** The Stack Pointer originally contains the value 0BH. Internal RAM locations 0AH and 0BH contain the values 23H and 01H, respectively. The instruction,

RET

will leave the Stack Pointer equal to the value 09H. Program execution will continue at location 0123H.

**Bytes:** 1**Cycles:** 2**Encoding:**

0 0 1 0 0 0 1 0

**Operation:**

RET

 $(PC_{15-8}) \leftarrow ((SP))$  $(SP) \leftarrow (SP) - 1$  $(PC_{7-0}) \leftarrow ((SP))$  $(SP) \leftarrow (SP) - 1$ **RETI****Function:** Return from interrupt**Description:** RETI pops the high- and low-order bytes of the PC successively from the stack, and restores the interrupt logic to accept additional interrupts at the same priority level as the one just processed. The Stack Pointer is left decremented by two. No other registers are affected; the PSW is not automatically restored to its pre-interrupt status. Program execution continues at the resulting address, which is generally the instruction immediately after the point at which the interrupt request was detected. If a lower- or same-level interrupt has been pending when the RETI instruction is executed, that one instruction will be executed before the pending interrupt is processed.**Example:** The Stack Pointer originally contains the value 0BH. An interrupt was detected during the instruction ending at location 0122H. Internal RAM locations 0AH and 0BH contain the values 23H and 01H, respectively. The instruction,

RETI

will leave the Stack Pointer equal to 09H and return program execution to location 0123H.

**Bytes:** 1**Cycles:** 2**Encoding:**

0 0 1 1 0 0 1 0

**Operation:**

RETI

 $(PC_{15-8}) \leftarrow ((SP))$  $(SP) \leftarrow (SP) - 1$  $(PC_{7-0}) \leftarrow ((SP))$  $(SP) \leftarrow (SP) - 1$

## 80C51 Family

80C51 family programmer's guide  
and instruction set

## RL A

**Function:** Rotate Accumulator Left**Description:** The eight bits in the Accumulator are rotated one bit to the left. Bit 7 is rotated into the bit 0 position. No flags are affected.**Example:** The Accumulator holds the value 0C5H (11000101B). The instruction, RL A, leaves the Accumulator holding the value 8BH (10001011B) with the carry unaffected.**Bytes:** 1**Cycles:** 1**Encoding:**

0	0	1	0
---	---	---	---

0	0	1	1
---	---	---	---

**Operation:** RL  
 $(A_{n+1}) \leftarrow (A_n), n = 0 - 6$   
 $(A_0) \leftarrow (A_7)$ 

## RLC A

**Function:** Rotate Accumulator Left through the Carry flag**Description:** The eight bits in the Accumulator and the carry flag are together rotated one bit to the left. Bit 7 moves into the carry flag; the original state of the carry flag moves into the bit 0 position. No other flags are affected.**Example:** The Accumulator holds the value 0C5H (11000101B), and the carry is zero. The instruction, RLC A, leaves the Accumulator holding the value 8BH (10001010B) with the carry set.**Bytes:** 1**Cycles:** 1**Encoding:**

0	0	1	1
---	---	---	---

0	0	1	1
---	---	---	---

**Operation:** RLC  
 $(A_{n+1}) \leftarrow (A_n), n = 0 - 6$   
 $(A_0) \leftarrow (C)$   
 $(C) \leftarrow (A_7)$



80C51 Family

80C51 family programmer's guide  
and instruction set

RR A

**Function:** Rotate Accumulator Right

**Description:** The eight bits in the Accumulator are rotated one bit to the right. Bit 0 is rotated into the bit 7 position. No flags are affected.

**Example:** The Accumulator holds the value 0C5H (11000101B). The instruction, RR A leaves the Accumulator holding the value 0E2H (11100010B) with the carry unaffected.

**Bytes:** 1

**Cycles:** 1

**Encoding:** 0 0 0 0 0 0 1 1

**Operation:** RR  
 $(A_n) \leftarrow (A_{n+1}), n = 0 - 6$   
 $(A7) \leftarrow (A0)$

RRC A

**Function:** Rotate Accumulator Right through the Carry flag

**Description:** The eight bits in the Accumulator and the carry flag are together rotated one bit to the right. Bit 0 moves into the carry flag; the original state of the carry flag moves into the bit 7 position. No other flags are affected.

**Example:** The Accumulator holds the value 0C5H (11000101B), and the carry is zero. The instruction, RRC A leaves the Accumulator holding the value 62 (01100010B) with the carry set.

**Bytes:** 1

**Cycles:** 1

**Encoding:** 0 0 0 1 0 0 1 1

**Operation:** RRC  
 $(A_n) \leftarrow (A_{n+1}), n = 0 - 6$   
 $(A7) \leftarrow (C)$   
 $(C) \leftarrow (A0)$

## 80C51 Family

80C51 family programmer's guide  
and instruction set**SETB <bit>****Function:** Set Bit**Description:** SETB sets the indicated bit to one. SETB can operate on the carry flag or any directly addressable bit. No other flags are affected.**Example:** The carry flag is cleared. Output Port 1 has been written with the value 34H (00110100B). The instructions,

SETB C

SETB P1.0

will leave the carry flag set to 1 and change the data output on Port 1 to 35H (00110101B).

**SETB C****Bytes:** 1**Cycles:** 1**Encoding:**

1 1 0 1 0 0 1 1

**Operation:**

SETB

(C)  $\leftarrow 1$ **SETB bit****Bytes:** 2**Cycles:** 1**Encoding:**

1 1 0 1 0 0 1 0

bit address

**Operation:**

SETB

(bit)  $\leftarrow 1$ **SJMP rel****Function:** Short Jump**Description:** Program control branches unconditionally to the address indicated. The branch destination is computed by adding the signed displacement in the second instruction byte to the PC, after incrementing the PC twice. Therefore, the range of destinations allowed is from 128 bytes preceding this instruction to 127 bytes following it.**Example:** The label "RELADR" is assigned to an instruction at program memory location 0123H. The instruction, SJMP RELADR will assemble into location 0100H. After the instruction is executed, the PC will contain the value 0123H. (Note: Under the above conditions the instruction following SJMP will be at 102H. Therefore, the displacement byte of the instruction will be the relative offset (0123H-0102H) = 21H. Put another way, an SJMP with a displacement of 0FEH would be a one-instruction infinite loop.)**Bytes:** 2**Cycles:** 2**Encoding:**

1 0 0 0 0 0 0 0

rel. address

**Operation:**

SJMP

(PC)  $\leftarrow$  (PC) + 2(PC)  $\leftarrow$  (PC) + rel

80C51 Family

80C51 family programmer's guide  
and instruction set

SUBB A, <src-byte>

**Function:** Subtract with borrow

**Description:** SUBB subtracts the indicated variable and the carry flag together from the Accumulator, leaving the result in the Accumulator. SUBB sets the carry (borrow) flag if a borrow is needed for bit 7, and clears C otherwise. (If C was set *before* executing a SUBB instruction, this indicates that a borrow was needed for the previous step in a multiple precision subtraction, so the carry is subtracted from the Accumulator along with the source operand.) AC is set if a borrow is needed for bit 3, and cleared otherwise. OV is set if a borrow is needed into bit 6, but not into bit 7, or into bit 7, but not bit 6.

When subtracting signed integers OV indicates a negative number produced when a negative value is subtracted from a positive value, or a positive result when a positive number is subtracted from a negative number.

The source operand allows four addressing modes: register, direct, register-indirect, or immediate.

**Example:** The Accumulator holds 0C9H (11001001B), register 2 holds 54H (01010100B), and the carry flag is set. The instruction,

SUBB A,R2

will leave the value 74H (01110100B) in the Accumulator, with the carry flag and AC cleared but OV set.

Notice that 0C9H minus 54H is 75H. The difference between this and the above result is due to the carry (borrow) flag being set before the operation. If the state of the carry is not known before starting a single or multiple-precision subtraction, it should be explicitly cleared by a CLR C instruction

SUBB A,Rn

**Bytes:** 1

**Cycles:** 1

**Encoding:** 1 0 0 1 1 r r r

**Operation:** SUBB  
 $(A) \leftarrow (A) - (C) - (R_n)$

SUBB A,direct

**Bytes:** 2

**Cycles:** 1

**Encoding:** 1 0 0 1 0 1 0 1 direct address

**Operation:** SUBB  
 $(A) \leftarrow (A) - (C) - (\text{direct})$

SUBB A,@Ri

**Bytes:** 1

**Cycles:** 1

**Encoding:** 1 0 0 1 0 1 1 i

**Operation:** SUBB  
 $(A) \leftarrow (A) - (C) - (R_i)$

SUBB A,#data

**Bytes:** 2

**Cycles:** 1

**Encoding:** 1 0 0 1 0 1 0 0 immediate data

**Operation:** SUBB  
 $(A) \leftarrow (A) - (C) - (\#data)$

## 80C51 Family

80C51 family programmer's guide  
and instruction set**SWAP A****Function:** Swap nibbles within the Accumulator**Description:** SWAP A interchanges the low- and high-order nibbles (four-bit fields) of the Accumulator (bits 3-0 and bits 7-4). The operation can also be thought of as a four-bit rotate instruction. No flags are affected.**Example:** The Accumulator holds the value 0C5H (11000101B). The instruction,

SWAP A

leaves the Accumulator holding the value 5CH (01011100B).

**Bytes:** 1**Cycles:** 1**Encoding:**

1	1	0	0	0	1	0	0
---	---	---	---	---	---	---	---

**Operation:** SWAP $(A_{3-0}) \leftrightarrow (A_{7-4})$ **XCH A,<byte>****Function:** Exchange Accumulator with byte variable**Description:** XCH loads the Accumulator with the contents of the indicated variable, at the same time writing the original Accumulator contents to the indicated variable. The source/destination operand can use register, direct, or register-indirect addressing.**Example:** R0 contains the address 20H. The Accumulator holds the value 3FH (00111111B). Internal RAM location 20H holds the value 75H (01110101B). The instruction,

XCH A @R0

will leave the RAM location 20H holding the values 3FH (00111111B) and 75H (01110101B) in the Accumulator.

**XCH A,Rn****Bytes:** 1**Cycles:** 1**Encoding:**

1	1	0	0	1	r	r	r
---	---	---	---	---	---	---	---

**Operation:** XCH $(A) \leftrightarrow (R_n)$ **XCH A,direct****Bytes:** 2**Cycles:** 1**Encoding:**

1	1	0	0	0	1	0	1
---	---	---	---	---	---	---	---

direct address

**Operation:** XCH $(A) \leftrightarrow (\text{direct})$ **XCH A,@Ri****Bytes:** 1**Cycles:** 1**Encoding:**

1	1	0	0	0	1	1	i
---	---	---	---	---	---	---	---

**Operation:** XCH $(A) \leftrightarrow ((R_i))$

## 80C51 Family

80C51 family programmer's guide  
and instruction set**XCHD A,@Ri****Function:** Exchange Digit**Description:** XCHD exchanges the low-order nibble of the Accumulator (bits 3-0), generally representing a hexadecimal or BCD digit, with that of the internal RAM location indirectly addressed by the specified register. The high-order nibbles (bits 7-4) of each register are not affected. No flags are affected.**Example:** R0 contains the address 20H. The Accumulator holds the value 36H (00110110B). Internal RAM location 20H holds the value 75H (01110101B). The instruction, XCHD A,@R0 will leave RAM location 20H holding the value 76H (01110110B) and 35H (00110101B) in the Accumulator.**Bytes:** 1**Cycles:** 1**Encoding:** 1 1 0 1 0 1 1 i  
**Operation:** XCHD  
(A<sub>3-0</sub>)  $\leftrightarrow$  ((Ri<sub>3-0</sub>))**XRL <dest-byte>,<src-byte>****Function:** Logical Exclusive-OR for byte variables**Description:** XRL performs the bitwise logical Exclusive-OR operation between the indicated variables, storing the results in the destination. No flags are affected.

The two operands allow six addressing mode combinations. When the destination is the Accumulator, the source can use register, direct, register-indirect, or immediate addressing; when the destination is a direct address, the source can be the Accumulator or immediate data.

(Note: When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, not the input pins.)

**Example:** If the Accumulator holds 0C3H (11000011B) and register 0 holds 0AAH (10101010B) then the instruction, XRL A,R0 will leave the Accumulator holding the value 69H (01101001B).

When the destination is a directly addressed byte, this instruction can complement combinations of bits in any RAM location or hardware register. The pattern of bits to be complemented is then determined by a mask byte, either a constant contained in the instruction or a variable computed in the Accumulator at run-time. The instruction,

XRL P1,#00110001B

will complement bits 5, 4, and 0 of output Port 1.

## 80C51 Family

80C51 family programmer's guide  
and instruction set**XRL A,Rn****Bytes:** 1**Cycles:** 1**Encoding:**

0 1 1 0 1 r r r

**Operation:** XRL $(A) \leftarrow (A) \vee (R_n)$ **XRL A,direct****Bytes:** 2**Cycles:** 1**Encoding:**

0 1 1 0 0 1 0 1

direct address

**Operation:** XRL $(A) \leftarrow (A) \vee (\text{direct})$ **XRL A,@Ri****Bytes:** 1**Cycles:** 1**Encoding:**

0 1 1 0 0 1 1 i

**Operation:** XRL $(A) \leftarrow (A) \vee (R_i)$ **XRL A,#data****Bytes:** 2**Cycles:** 1**Encoding:**

0 1 1 0 0 1 0 0

immediate data

**Operation:** XRL $(A) \leftarrow (A) \vee \#data$ **XRL direct,A****Bytes:** 2**Cycles:** 1**Encoding:**

0 1 1 0 0 0 1 0

direct address

**Operation:** XRL $(\text{direct}) \leftarrow (\text{direct}) \vee (A)$ **XRL direct,#data****Bytes:** 3**Cycles:** 2**Encoding:**

0 1 1 0 0 0 1 1

direct address

immediate data

**Operation:** XRL $(\text{direct}) \leftarrow (\text{direct}) \vee \#data$



## 80C51 Family

## 80C51 family EPROM products

## EPROM PRODUCTS

Most of the 80C51 derivative products offered by Philips are supported with an EPROM version.

All EPROM products are available in both windowed DIP and OTP package configurations. The windowed DIP package allows the EPROM to be erased under a strong UV light source, making program development easier and faster. The OTP (One Time Programmable) version cannot be erased because there is no window through which the die could be exposed to UV light. While the EPROM can only be programmed once in the OTP package, the part costs less than in windowed DIP and therefore offers an advantage for those not desiring to use the masked ROM version of the part.

The EPROM products are fully supported on the industry standard EPROM programmers.

## Programming the 87C51

The setup for programming the microcontroller is shown in Figure 1. Note that the part is running with a 4 to 6 MHz oscillator. The clock must be running because the device is executing internal address and program data transfers during the programming.

To program the 87C51, the address of the EPROM location to be programmed is applied to ports 1 and 2 as shown in Figure 1. The code byte to be programmed into this location is applied to port 0. RST, PSEN, and the pins of ports 2 and 3 specified in Table 1 are held at the "Program Code Data" levels specified in the table. The ALE/PROG is then pulsed low 25 times to program the addressed location.

## Encryption Table

The encryption table is a feature of the 87C51, and its derivatives, that protects the code from being easily read by anyone other than the programmer. The encryption table is 16 to 64 bytes of code, depending on the microcontroller, that are exclusive NORed with the program code data as it is read out. The first byte is XNORed with the first location read, the second with the second read, etc. through the sixteenth byte read. The seventeenth byte is XNORed with the first byte of the encryption table, the eighteenth with the second, etc. and on in sixteen-byte groups.

After the Encryption table has been programmed the user has to know its

contents in order to correctly decode the program code data. The encryption table itself cannot be read out.

The encryption table is programmed in the same manner as the program memory, but using the "Pgm Encryption Table" levels specified in Table 1. After the encryption table is programmed, verification cycles will produce only encrypted information.

## Security Bit

There are two security bits on the 87C51 that, when set, prevent the program data memory from being read out or programmed further. To program the security bits, repeat the programming sequence using the "Pgm Security Bit" levels specified in Table 1.

After the first security bit is programmed, further programming of the code memory or the encryption table is disabled. The other security bit can of course still be programmed. With only security bit one programmed, the memory can still be read out for program verification. After the second security bit is programmed, it is no longer possible to read out (verify) the program memory.

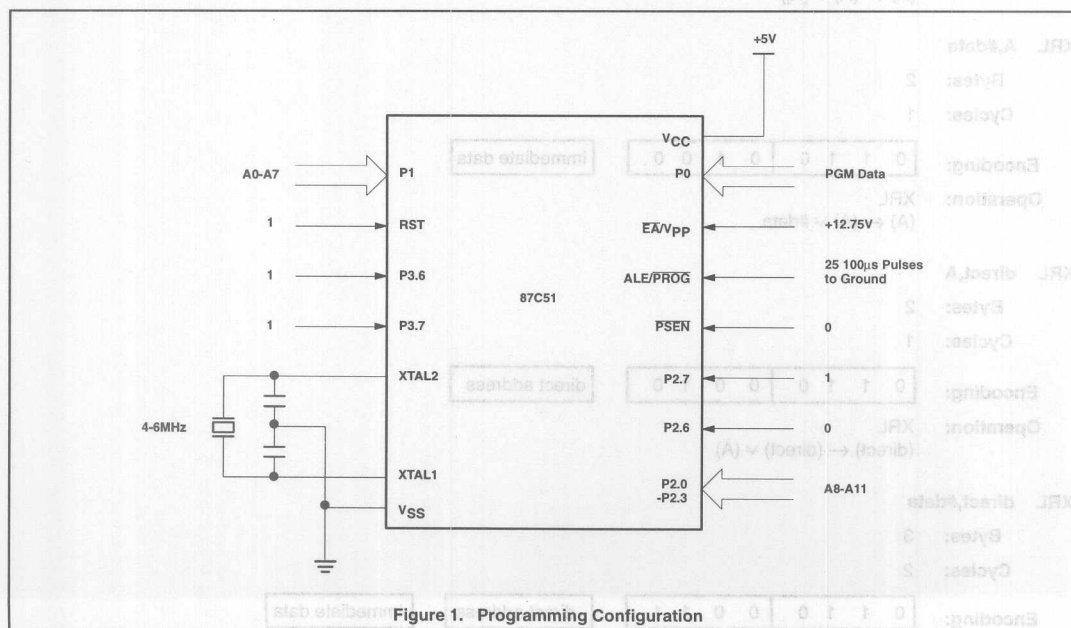


Figure 1. Programming Configuration

## 80C51 Family

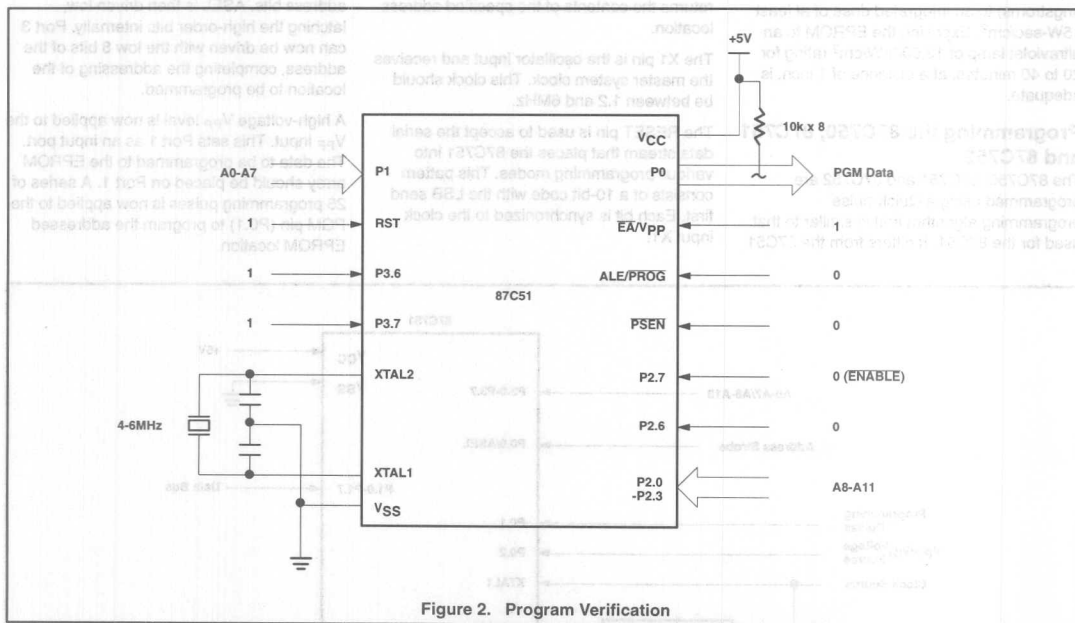
## 80C51 family EPROM products

### Table 1. EPROM Programming Modes

MODE	RST	PSEN	ALE/PROG	EA/V <sub>PP</sub>	P2.7	P2.6	P3.7	P3.6
Read signature	1	0	1	1	0	0	0	0
Program code data	1	0	0*	V <sub>PP</sub>	1	0	1	1
Verify code data	1	0	1	1	0	0	1	1
Pgm encryption table	1	0	0*	V <sub>PP</sub>	1	0	1	0
Pgm security bit 1	1	0	0*	V <sub>PP</sub>	1	1	1	1
Pgm security bit 2	1	0	0*	V <sub>PP</sub>	1	1	0	0

**NOTES:**

1. "0" = valid low for that pin, "1" = valid high for that pin.
2.  $V_{PP} = 12.75 \pm 0.25V$ .
3.  $V_{CC} = 5V \pm 10\%$  during programming and verification.
- \* ALE/PROG receives 25 programming pulses while  $V_{PP}$  is held at 12.75V. Each programming pulse is low for 100ms ( $\pm 10\mu s$ ) and high for a minimum of 10 $\mu s$ .



## Program Verification

If security bit 2 has not been programmed the on-chip program memory can be read out for program verification. To verify the contents of the program memory, the address of the location to be read is applied to ports 1 and 2 as shown in Figure 2. The other pins are held at the "Verify Code Data" levels indicated in Table 1. The contents of the addressed location will appear on port 0. For this operation external pull-ups are required on port 0 as shown in Figure 2. Note that if the encryption table has been programmed the data presented at port 0 will be the exclusive

NOR of the program byte with a byte from the encryption table.

### Signature Bytes

The 87C51 contains two signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes identify the device as an 87C51 manufactured by Philips.

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. The values are:

(030H) = 15H indicates the part is made by Philips

(031H) =	90H 87C451	9DH 87C524
	92H 87C51	9EH 87C598
	94H 87C552	B0H 87C575
	96H 87C550	B1H 87C51FA
	97H 87C52	B2H 87C51FB
	99H 87C654	B3H 87C51FC
	/87C652	B5H 89CE558
	9AH 80C52	B6H 87C576
	9BH 87C528	BBH 87C504
	9CH 87C592	4BH 87C055

## 80C51 Family

## 80C51 family EPROM products

**EPROM Erasure**

Erasure of the EPROM occurs when the chip is exposed to light with wavelengths shorter than 4000 angstroms. Sunlight and fluorescent lighting have wavelengths in this range, so exposure to these light sources over an extended period of time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent erasure. It is recommended, for this reason, that an opaque label be placed over the window. If the part is subject to elevated temperatures or an environment where solvents are used, Kapton tape (Fluorglas part number 2345-5 or its equivalent) can be used.

The recommended erasure procedure is to expose the chip to ultraviolet light (at 2537 angstroms) to an integrated dose of at least 15W-sec/cm<sup>2</sup>. Exposing the EPROM to an ultraviolet lamp of 12,000µW/cm<sup>2</sup> rating for 20 to 40 minutes, at a distance of 1 inch, is adequate.

**Programming the 87C750, 87C751 and 87C752**

The 87C750, 87C751 and 87C752 are programmed using a Quick-pulse programming algorithm that is similar to that used for the 87C51. It differs from the 87C51

in that a serial data stream is used to place the 87C751 in the programming mode.

Figure 3 shows a block diagram of the programming configuration for the 87C751. Port pin P0.2 is used for the programming voltage supply input (V<sub>PP</sub> signal). Port pin P0.1 is used for the program (PGM) signal.

Port 3 accepts the address input for the EPROM location to be programmed. Both the high and low components of the eleven-bit address are presented to the part through port 3. Multiplexing of the address components is performed using ASEL (P0.0).

Port 1 is used as a bidirectional data bus during programming and verify operations. During the programming mode, it accepts the byte to be programmed. In the verify mode, it returns the contents of the specified address location.

The X1 pin is the oscillator input and receives the master system clock. This clock should be between 1.2 and 6MHz.

The RESET pin is used to accept the serial data stream that places the 87C751 into various programming modes. This pattern consists of a 10-bit code with the LSB sent first. Each bit is synchronized to the clock input X1.

To program the 87C751 the part must be put into the programming mode by presenting the proper serial code (see Table 2) to the RESET pin. To do this RESET should be held high for at least two machine cycles. Port pins P0.1 and P0.2 will be at VOH as a result of this, but they must be driven high prior to sending the serial data stream on the RESET pin. The serial data bits can now be transmitted over the RESET pin placing the 87C751 into one of the programming modes. Following the transmission of the last data bit, the reset pin should be held low.

Next the address information for the location to be programmed is placed on Port 3 and ASEL is used to perform the address multiplexing. ASEL should be driven high and then Port 3 driven with the high-order address bits. ASEL is then driven low, latching the high-order bits internally. Port 3 can now be driven with the low 8 bits of the address, completing the addressing of the location to be programmed.

A high-voltage V<sub>PP</sub> level is now applied to the V<sub>PP</sub> input. This sets Port 1 as an input port. The data to be programmed to the EPROM array should be placed on Port 1. A series of 25 programming pulses is now applied to the PGM pin (P0.1) to program the addressed EPROM location.

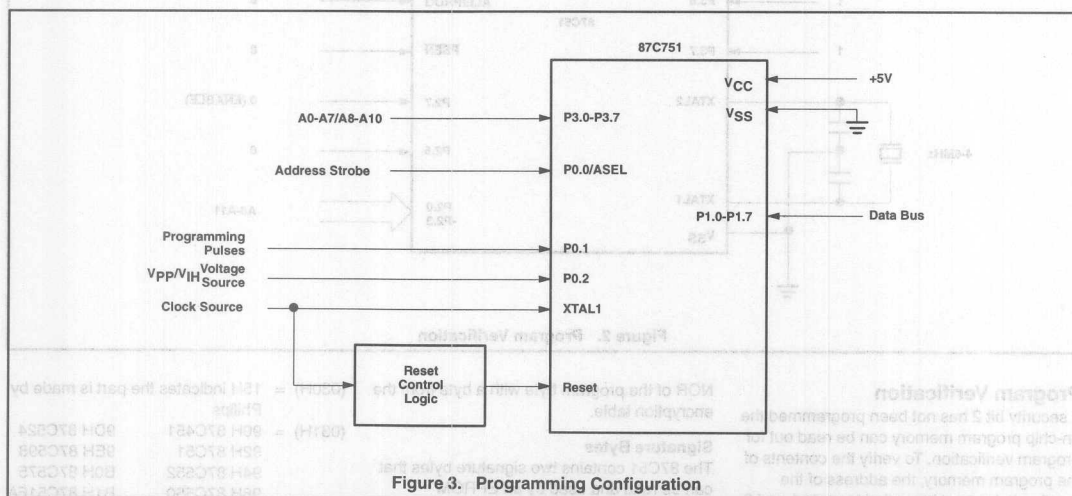


Figure 3. Programming Configuration

## 80C51 Family

## 80C51 family EPROM products

**Table 2. 87C750, 87C751, and 87C752 Serial Codes**

OPERATION	SERIAL CODE	P0.1 (PGM/)	P0.2 (V <sub>PP</sub> )
Program user EPROM	296H	—*	V <sub>PP</sub>
Verify user EPROM	296H	V <sub>IH</sub>	V <sub>IH</sub>
Program key EPROM	292H	—*	V <sub>PP</sub>
Verify key EPROM	292H	V <sub>IH</sub>	V <sub>IH</sub>
Program security bit 1	29AH	—*	V <sub>PP</sub>
Program security bit 2	298H	—*	V <sub>PP</sub>
Verify security bits	29AH	V <sub>IH</sub>	V <sub>IH</sub>
Read signature bytes	294H	V <sub>IH</sub>	V <sub>IH</sub>

**NOTE:**

\* Pulsed from V<sub>IH</sub> to V<sub>IL</sub> and returned to V<sub>IH</sub>.

**Program Verification**

The EPROM array can be verified by placing the part in the programming mode as described above and forcing the V<sub>PP</sub> pin to the V<sub>OH</sub> level. Four machine cycles after addressing a location the contents of the addressed location will appear on Port 1.

**87C750, 87C751 and 87C752 Signature Bytes**

The signature bytes for the 87C750, 87C751 and 87C752 are read differently and are in different locations than those on the 87C51. Due to its reduced pin count, the part has to be put into "Signature Byte Read Mode" by

placing a 10-bit serial data stream on the Reset pin. The proper code and the conditions of P0.1 and P0.2, for this mode, are shown in Table 2.

Once the part has been placed into the Signature Byte Read Mode, the signature bytes can be read by the same procedure as a normal verification of locations 01EH and 01FH. The values are:

01EH = 15H indicates the part is made by Philips  
 01FH = 91H - 87C751  
 01FH = 95H - 87C752

**Programming Features**

The 87C751 has all of the special programming features incorporated within its EPROM array that the 87C51 has. It has an encryption key table and two security bits. These function exactly as they do in the 87C51. They are programmed or verified by sending the proper code over the RESET pin (see Table 2) and then following the 87C751 programming procedure as described previously.

**Erase Characteristics**

The erasure procedure is exactly the same as that described for the 87C51.



# Section 3

## 80C51 Family Derivatives

### 80C51-Based 8-Bit Microcontrollers

#### CONTENTS

<b>80C31/80C51/87C51</b>	CMOS single-chip 8-bit microcontroller	135
<b>83C51FA/87C51FA</b>	CMOS single-chip 8-bit microcontroller	153
<b>83C51FB/87C51FB</b>	CMOS single-chip 8-bit microcontrollers	168
<b>83C51FC/87C51FC</b>	CMOS single-chip 8-bit microcontrollers	194
<b>80CL31/80CL51</b>	Low-voltage single-chip 8-bit microcontrollers	220
<b>83L51FA/87L51FA</b>	CMOS single-chip 3.0V 8-bit microcontroller	254
<b>83L51FB/87L51FB</b>	CMOS single-chip 3.0V 8-bit microcontroller	269
<b>80C32/80C52/87C52</b>	CMOS single-chip 8-bit microcontrollers	284
<b>83C055/87C055</b>	Microcontroller for television and video (MTV)	304
<b>P83CL168/167/268/267</b>	Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function	323
<b>80CL410/83CL410</b>	Low voltage/low power single-chip 8-bit microcontroller with I <sup>2</sup> C	382
<b>80C451/83C451/87C451</b>	CMOS single-chip 8-bit microcontroller	405
<b>83C504/87C504</b>	CMOS single-chip 8-bit microcontroller	424
<b>P83C524</b>	8-bit microcontroller	440
<b>87C524</b>	CMOS single-chip 8-bit microcontroller	460
<b>80C528/83C528</b>	CMOS single-chip 8-bit microcontroller	478
<b>87C528</b>	CMOS single-chip 8-bit microcontroller	495
<b>P8xCE528</b>	8-bit microcontroller with EMC and EEPROM	382
<b>80C550/83C550/87C550</b>	CMOS single-chip 8-bit microcontroller with A/D and watchdog timer	554
<b>8XC552/562 overview</b>		575
<b>8XC552 OVERVIEW</b>		575
<b>83C562 OVERVIEW</b>		575
<b>Differences From the 80C51</b>		575
Program Memory		575
Data Memory		575
Special Function Registers		575
Timer T2		576
Timer T3, The Watchdog Timer		582
Serial I/O		583
Reset Circuitry		616
Interrupts		617
I/O Port Structure		621
Port 1 Operation		621
Port 5 Operation		621
Pulse Width Modulated Outputs		621
Analog-to-Digital Converter		621
Power Reduction Modes		627
Memory Organization		629
<b>80C552/83C552</b>	Single-chip 8-bit microcontroller with 10-bit A/D, capture/compare timer, high-speed outputs, PWM	634
<b>87C552</b>	Single-chip 8-bit microcontroller with 10-bit A/D, capture/compare timer, high-speed outputs, PWM	654
<b>80CE558/83CE558/89CE558</b>	Single-chip 8-bit microcontroller	674
<b>80C562/83C562</b>	Single-chip 8-bit microcontroller with 8-bit A/D, capture/compare timer, high-speed outputs, PWM	727



80C51 Family Derivatives

Section 3

CONTENTS (Continued)

80C575/83C575/87C575	CMOS single-chip 8-bit microcontroller	740
83C576/87C576	CMOS single-chip 8-bit microcontroller	774
80CL580/83CL580	Low-voltage single-chip 8-bit microcontroller	811
80C592/83C592/87C592	Single-chip 8-bit microcontroller with CAN controller	851
P8xC598	8-bit microcontroller with on-chip CAN	889
80C652/83C652	CMOS single-chip 8-bit microcontroller	916
87C652	CMOS single-chip 8-bit microcontroller	934
83C654	CMOS single-chip 8-bit microcontroller	954
87C654	CMOS single-chip 8-bit microcontroller	972
80CE654/83CE654	CMOS single-chip 8-bit microcontroller with Electromagnetic Compatibility improvements	992
83C750/87C750	CMOS single-chip 8-bit microcontrollers	1006
83C751/87C751	CMOS single-chip 8-bit microcontroller	1016
83C752/87C752	CMOS single-chip 8-bit microcontroller with A/D, PWM	1034
83CL781/83CL782	Low-voltage single-chip 8-bit microcontrollers	1052
80C851/83C851	CMOS single-chip 8-bit microcontroller with on-chip EEPROM	1085
83C852	Secured 8-bit microcontroller	1100

# CMOS single-chip 8-bit microcontroller 80C31/80C51/87C51

## DESCRIPTION

The Philips 80C31/80C51/87C51 is a high-performance microcontroller fabricated with Philips high-density CMOS technology. The CMOS 80C51 is functionally compatible with the NMOS 8031/8051 microcontrollers. The Philips CMOS technology combines the high speed and density characteristics of HMOS with the low power attributes of CMOS. Philips epitaxial substrate minimizes latch-up sensitivity.

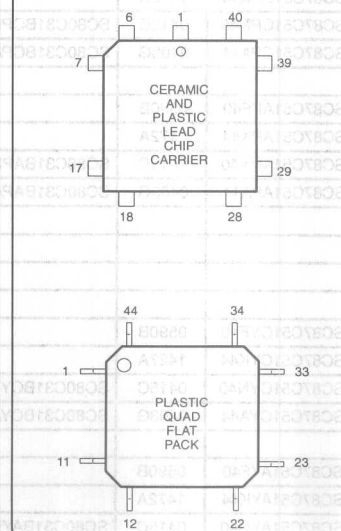
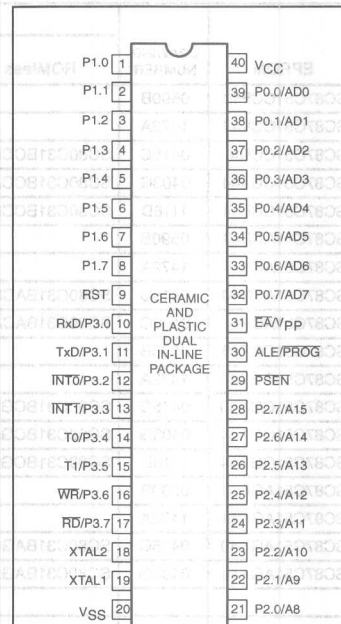
The 80C51 contains a 4k × 8 ROM (80C51) EPROM (87C51), a 128 × 8 RAM, 32 I/O lines, two 16-bit counter/timers, a five-source, two-priority level nested interrupt structure, a serial I/O port for either multi-processor communications, I/O expansion or full duplex UART, and on-chip oscillator and clock circuits.

In addition, the device has two software selectable modes of power reduction—idle mode and power-down mode. The idle mode freezes the CPU while allowing the RAM, timers, serial port, and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.

## FEATURES

- 8031/8051 compatible
  - 4k × 8 ROM (80C51)
  - 4k × 8 EPROM (87C51)
  - ROMless (80C31)
  - 128 × 8 RAM
  - Two 16-bit counter/timers
  - Full duplex serial channel
  - Boolean processor
- Memory addressing capability
  - 64k ROM and 64k RAM
- Power control modes:
  - Idle mode
  - Power-down mode
- CMOS and TTL compatible
- Five speed ranges at  $V_{CC} = 5V$ 
  - 12MHz
  - 16MHz
  - 24MHz
  - 30MHz
  - 33MHz
- Five package styles
- Extended temperature ranges
- OTP package available

## PIN CONFIGURATIONS



SEE PAGE 138 FOR QFP AND LCC PIN FUNCTIONS.

## CMOS single-chip 8-bit microcontroller

80C31/80C51/87C51

## ORDERING INFORMATION

PHILIPS NORTH AMERICA						
EPROM	DRAWING NUMBER	ROMless	ROM	DRAWING NUMBER	TEMPERATURE RANGE °C AND PACKAGE <sup>1</sup>	Freq MHz
SC87C51CCF40	0590B				0 to +70, Ceramic Dual In-line Package, UV	3.5 to 12
SC87C51CCK44	1472A				0 to +70, Ceramic Leaded Chip Carrier, UV	3.5 to 12
SC87C51CCN40	0415C	SC80C31BCCN40	SC80C51BCCN40	0415C	0 to +70, Plastic Dual In-line Package, OTP	3.5 to 12
SC87C51CCA44	0403G	SC80C31BCCA44	SC80C51BCCA44	0403G	0 to +70, Plastic Leaded Chip Carrier, OTP	3.5 to 12
SC87C51CCB44	1118D	SC80C31BCCB44	SC80C51BCCB44	1118D	0 to +70, Plastic Quad Flat Pack, OTP	3.5 to 12
SC87C51ACF40	0590B				-40 to +85, Ceramic Dual In-line Package, UV	3.5 to 12
SC87C51ACK44	1472A				-40 to +85, Ceramic Leaded Chip Carrier, UV	3.5 to 12
SC87C51ACN40	0415C	SC80C31BACN40	SC80C51BACN40	0415C	-40 to +85, Plastic Dual In-line Package, OTP	3.5 to 12
SC87C51ACA44	0403G	SC80C31BACA44	SC80C51BACA44	0403G	-40 to +85, Plastic Leaded Chip Carrier, OTP	3.5 to 12
SC87C51CGF40	0590B				0 to +70, Ceramic Dual In-line Package, UV	3.5 to 16
SC87C51CGK44	1472A				0 to +70, Ceramic Leaded Chip Carrier, UV	3.5 to 16
SC87C51CGN40	0415C	SC80C31BCGN40	SC80C51BCGN40	0415C	0 to +70, Plastic Dual In-line Package, OTP	3.5 to 16
SC87C51CGA44	0403G	SC80C31BCGA44	SC80C51BCGA44	0403G	0 to +70, Plastic Leaded Chip Carrier, OTP	3.5 to 16
SC87C51CGB44	1118D	SC80C31BCGB44	SC80C51BCGB44	1118D	0 to +70, Plastic Quad Flat Pack, OTP	3.5 to 16
SC87C51AGF40	0590B				-40 to +85, Ceramic Dual In-line Package, UV	3.5 to 16
SC87C51AGK44	1472A				-40 to +85, Ceramic Leaded Chip Carrier, UV	3.5 to 16
SC87C51AGN40	0415C	SC80C31BAGN40	SC80C51BAGN40	0415C	-40 to +85, Plastic Dual In-line Package, OTP	3.5 to 16
SC87C51AGA44	0403G	SC80C31BAGA44	SC80C51BAGA44	0403G	-40 to +85, Plastic Leaded Chip Carrier, OTP	3.5 to 16
SC87C51CPF40	0590B				0 to +70, Ceramic Dual In-line Package, UV	3.5 to 24
SC87C51CPK44	1472A				0 to +70, Ceramic Leaded Chip Carrier, UV	3.5 to 24
SC87C51CPN40	0415C	SC80C31BCPN40	SC80C51BCPN40	0415C	0 to +70, Plastic Dual In-line Package, OTP	3.5 to 24
SC87C51CPA44	0403G	SC80C31BCPA44	SC80C51BCPA44	0403G	0 to +70, Plastic Leaded Chip Carrier, OTP	3.5 to 24
SC87C51APF40	0590B				-40 to +85, Ceramic Dual In-line Package, UV	
SC87C51APK44	1472A				-40 to +85, Ceramic Leaded Chip Carrier, UV	3.5 to 24
SC87C51APN40	0415C	SC80C31BAPN40	SC80C51BAPN40	0415C	-40 to +85, Plastic Dual In-line Package, OTP	3.5 to 24
SC87C51APA44	0403G	SC80C31BAPA44	SC80C51BAPA44	0403G	-40 to +85, Plastic Leaded Chip Carrier, OTP	3.5 to 24
SC87C51CYF40	0590B				0 to +70, Ceramic Dual In-line Package, UV	3.5 to 33
SC87C51CYK44	1472A				0 to +70, Ceramic Leaded Chip Carrier, UV	3.5 to 33
SC87C51CYN40	0415C	SC80C31BCYN40	SC80C51BCYN40	0415C	0 to +70, Plastic Dual In-line Package, OTP	3.5 to 33
SC87C51CYA44	0403G	SC80C31BCYA44	SC80C51BCYA44	0403G	0 to +70, Plastic Leaded Chip Carrier, OTP	3.5 to 33
SC87C51AYF40	0590B				-40 to +85, Ceramic Dual In-line Package, UV	3.5 to 33
SC87C51AYK44	1472A				-40 to +85, Ceramic Leaded Chip Carrier, UV	3.5 to 33
SC87C51AYN40	0415C	SC80C31BAYN40	SC80C51BAYN40	0415C	-40 to +85, Plastic Dual In-line Package, OTP	3.5 to 33
SC87C51AYA44	0403G	SC80C31BAYA44	SC80C51BAYA44	0403G	-40 to +85, Plastic Leaded Chip Carrier, OTP	3.5 to 33

1. OTP = One Time Programmable EPROM. UV = UV Erasable EPROM

2. SOT311 replaced by SOT307-2.

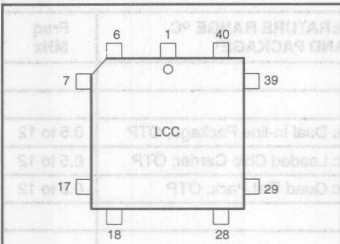
## CMOS single-chip 8-bit microcontroller 80C31/80C51/87C51

## ORDERING INFORMATION

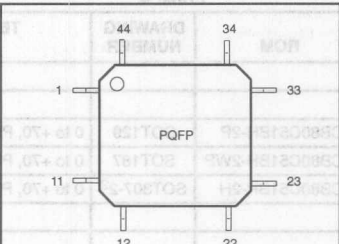
PHILIPS					
ROMless (ORDER NUMBER)	ROMless (MARKING NUMBER)	ROM	DRAWING NUMBER	TEMPERATURE RANGE °C AND PACKAGE <sup>1</sup>	Freq MHz
PCB80C31-2 N	PCB80C31BH2-12P	PCB80C51BH-2P	SOT129	0 to +70, Plastic Dual In-line Package, OTP	0.5 to 12
PCB80C31-2 A	PCB80C31BH2-12WP	PCB80C51BH-2WP	SOT187	0 to +70, Plastic Leaded Chip Carrier, OTP	0.5 to 12
	PCB80C31BH2-12H	PCB80C51BH-2H	SOT307-2 <sup>2</sup>	0 to +70, Plastic Quad Flat Pack, OTP	0.5 to 12
PCB80C31-3 N	PCB80C31BH3-16P	PCB80C51BH-3P	SOT129	0 to +70, Plastic Dual In-line Package, OTP	1.2 to 16
PCB80C31-3 A	PCB80C31BH3-16WP	PCB80C51BH-3WP	SOT187	0 to +70, Plastic Leaded Chip Carrier, OTP	1.2 to 16
	PCB80C31BH3-16H	PCB80C51BH-3H	SOT307-2 <sup>2</sup>	0 to +70, Plastic Quad Flat Pack, OTP	1.2 to 16
PCF80C31-3 N	PCF80C31BH3-16P	PCF80C51BH-3P	SOT129	-40 to +85, Plastic Dual In-line Package, OTP	1.2 to 16
PCF80C31-3 A	PCF80C31BH3-16WP	PCF80C51BH-3WP	SOT187	-40 to +85, Plastic Leaded Chip Carrier, OTP	1.2 to 16
	PCF80C31BH3-16H	PCF80C51BH-3H	SOT307-2 <sup>2</sup>	-40 to +85, Plastic Quad Flat Pack, OTP	1.2 to 16
	PCA80C31BH3-16P	PCA80C51BH-3P	SOT129	-40 to +125, Plastic Dual In-line Package	1.2 to 16
	PCA80C31BH3-16WP	PCA80C51BH-3WP	SOT187	-40 to +125, Plastic Leaded Chip Carrier	1.2 to 16
PCB80C31-4 N	PCB80C31BH4-24P	PCB80C51BH-4P	SOT129	0 to +70, Plastic Dual In-line Package, OTP	1.2 to 24
PCB80C31-4 A	PCB80C31BH4-24WP	PCB80C51BH-4WP	SOT187	0 to +70, Plastic Leaded Chip Carrier, OTP	1.2 to 24
	PCB80C31BH4-24H	PCB80C51BH-4H	SOT307-2 <sup>2</sup>	0 to +70, Plastic Quad Flat Pack, OTP	1.2 to 24
PCF80C31-4 N	PCF80C31BH4-24P	PCF80C51BH-4P	SOT129	-40 to +85, Plastic Dual In-line Package, OTP	1.2 to 24
PCF80C31-4 A	PCF80C31BH4-24WP	PCF80C51BH-4WP	SOT187	-40 to +85, Plastic Leaded Chip Carrier, OTP	1.2 to 24
	PCF80C31BH4-24H	PCF80C51BH-4H	SOT307-2 <sup>2</sup>	-40 to +85, Plastic Leaded Chip Carrier, OTP	1.2 to 24
PCB80C31-5 N	PCB80C31BH5-30P	PCB80C51BH-5P	SOT129	0 to +70, Plastic Dual In-line Package	1.2 to 33
PCB80C31-5 A	PCB80C31BH5-30WP	PCB80C51BH-5WP	SOT187	0 to +70, Plastic Leaded Chip Carrier	1.2 to 33
PCB80C31-5 B	PCB80C31BH5-30H	PCB80C51BH-5H	SOT307-2 <sup>2</sup>	0 to +70, Plastic Quad Flat Pack	1.2 to 33

CMOS single-chip 8-bit microcontroller 80C31/80C51/87C51

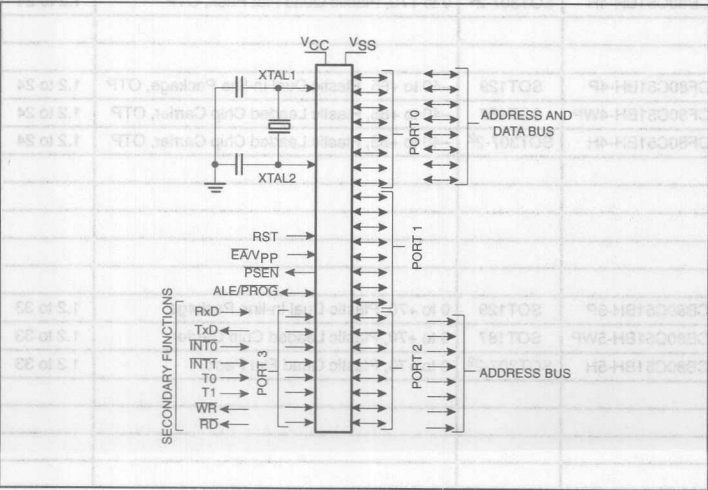
CERAMIC AND PLASTIC LEADED  
CHIP CARRIER PIN FUNCTIONS

			
Pin	Function	Pin	Function
1	NC	23	NC
2	P1.0	24	P2.0/A8
3	P1.1	25	P2.1/A9
4	P1.2	26	P2.2/A10
5	P1.3	27	P2.3/A11
6	P1.4	28	P2.4/A12
7	P1.5	29	P2.5/A13
8	P1.6	30	P2.6/A14
9	P1.7	31	P2.7/A15
10	RST	32	PSEN
11	P3.0/RxD	33	ALE/PROG
12	NC	34	NC
13	P3.1/TxD	35	EA/Vpp
14	P3.2/INT0	36	P0.7/AD7
15	P3.3/INT1	37	P0.6/AD6
16	P3.4/T0	38	P0.5/AD5
17	P3.5/T1	39	P0.4/AD4
18	P3.6/WR	40	P0.3/AD3
19	P3.7/RD	41	P0.2/AD2
20	XTAL2	42	P0.1/AD1
21	XTAL1	43	P0.0/AD0
22	VSS	44	VCC

PLASTIC QUAD FLAT PACK  
PIN FUNCTIONS

			
Pin	Function	Pin	Function
1	P1.5	23	P2.5/A13
2	P1.6	24	P2.6/A14
3	P1.7	25	P2.7/A15
4	RST	26	PSEN
5	P3.0/RxD	27	ALE/PROG
6	NC	28	NC
7	P3.1/TxD	29	EA/Vpp
8	P3.2/INT0	30	P0.7/AD7
9	P3.3/INT1	31	P0.6/AD6
10	P3.4/T0	32	P0.5/AD5
11	P3.5/T1	33	P0.4/AD4
12	P3.6/WR	34	P0.3/AD3
13	P3.7/RD	35	P0.2/AD2
14	XTAL2	36	P0.1/AD1
15	XTAL1	37	P0.0/AD0
16	VSS	38	VCC
17	NC	39	NC
18	P2.0/A8	40	P1.0
19	P2.1/A9	41	P1.1
20	P2.2/A10	42	P1.2
21	P2.3/A11	43	P1.3
22	P2.4/A12	44	P1.4

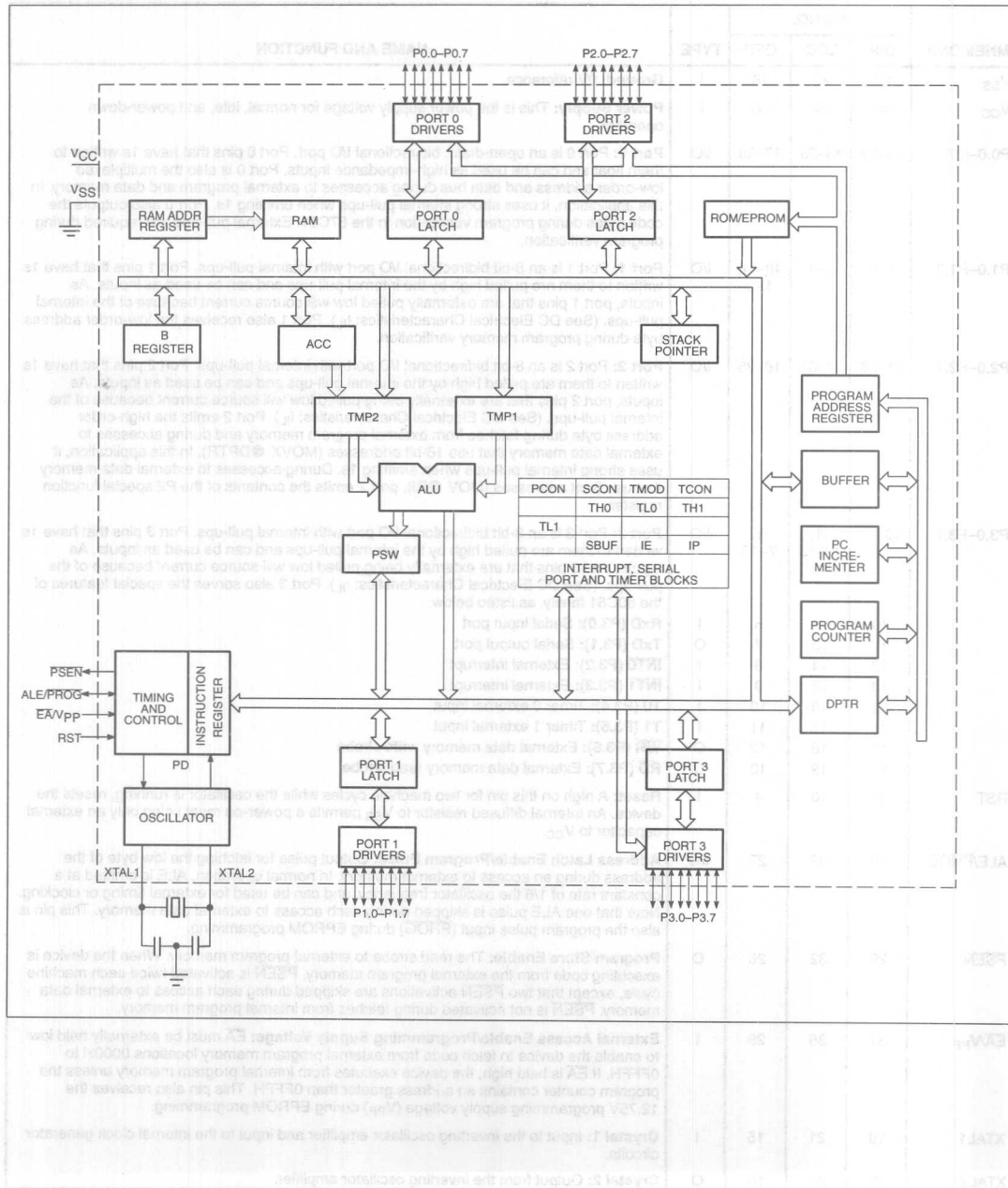
LOGIC SYMBOL





## CMOS single-chip 8-bit microcontroller 80C31/80C51/87C51

## BLOCK DIAGRAM





## CMOS single-chip 8-bit microcontroller 80C31/80C51/87C51

## PIN DESCRIPTION

MNEMONIC	PIN NO.			TYPE	NAME AND FUNCTION
	DIP	LCC	QFP		
V <sub>SS</sub>	20	22	16	I	<b>Ground:</b> 0V reference.
V <sub>CC</sub>	40	44	38	I	<b>Power Supply:</b> This is the power supply voltage for normal, idle, and power-down operation.
P0.0–P0.7	39–32	43–36	37–30	I/O	<b>Port 0:</b> Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s. Port 0 also outputs the code bytes during program verification in the 87C51. External pull-ups are required during program verification.
P1.0–P1.7	1–8	2–9	40–44, 1–3	I/O	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 1 also receives the low-order address byte during program memory verification.
P2.0–P2.7	21–28	24–31	18–25	I/O	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register.
P3.0–P3.7	10–17	11, 13–19	5, 7–13	I/O	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 3 also serves the special features of the 80C51 family, as listed below: <b>RxD (P3.0):</b> Serial input port <b>TxD (P3.1):</b> Serial output port <b>INT0 (P3.2):</b> External interrupt <b>INT1 (P3.3):</b> External interrupt <b>T0 (P3.4):</b> Timer 0 external input <b>T1 (P3.5):</b> Timer 1 external input <b>WR (P3.6):</b> External data memory write strobe <b>RD (P3.7):</b> External data memory read strobe
RST	9	10	4	I	<b>Reset:</b> A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to V <sub>SS</sub> permits a power-on reset using only an external capacitor to V <sub>CC</sub> .
ALE/PROG	30	33	27	I/O	<b>Address Latch Enable/Program Pulse:</b> Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory. This pin is also the program pulse input (PROG) during EPROM programming.
PSEN	29	32	26	O	<b>Program Store Enable:</b> The read strobe to external program memory. When the device is executing code from the external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory. PSEN is not activated during fetches from internal program memory.
E <sub>A</sub> V <sub>PP</sub>	31	35	29	I	<b>External Access Enable/Programming Supply Voltage:</b> E <sub>A</sub> must be externally held low to enable the device to fetch code from external program memory locations 0000H to 0FFFFH. If E <sub>A</sub> is held high, the device executes from internal program memory unless the program counter contains an address greater than 0FFFFH. This pin also receives the 12.75V programming supply voltage (V <sub>PP</sub> ) during EPROM programming.
XTAL1	19	21	15	I	<b>Crystal 1:</b> Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	20	14	O	<b>Crystal 2:</b> Output from the inverting oscillator amplifier.

# CMOS single-chip 8-bit microcontroller 80C31/80C51/87C51

## OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator, as shown in the logic symbol.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

## RESET

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. To insure a good power-up reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few milliseconds) plus two machine cycles.

## IDLE MODE

In idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle

mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

## POWER-DOWN MODE

In the power-down mode, the oscillator is stopped and the instruction to invoke power-down is the last instruction executed. Only the contents of the on-chip RAM are preserved. A hardware reset is the only way to terminate the power-down mode. The control bits for the reduced power modes are in the special function register PCON.

Table 1 shows the state of I/O ports during low current operating modes.

## ROM CODE SUBMISSION

When submitting ROM code for the 80C51, the following must be specified:

1. 4k byte user ROM data
2. 32 byte ROM encryption key (SC80C51 only)
3. ROM security bits (SC80C51 only).

ADDRESS	CONTENT	BIT(S)	COMMENT
0000H to 0FFFH	DATA	7:0	User ROM Data
1000H to 101FH	KEY	7:0	ROM Encryption Key
1020H	SEC	0	ROM Security Bit 1
1020H	SEC	1	ROM Security Bit 2

**Security Bit 1:** When programmed, this bit has two effects on masked ROM parts:

1. External MOVX is disabled, and
2. EA# is latched on Reset.

**Security Bit 2:** When programmed, this bit inhibits Verify User ROM.

**Table 1. External Pin Status During Idle and Power-Down Modes**

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

CMOS single-chip 8-bit microcontroller 80C31/80C51/87C51

Electrical Deviations from Commercial Specifications for Extended Temperature Range (87C51)  
DC and AC parameters not included here are the same as in the commercial temperature range table.

DC ELECTRICAL CHARACTERISTICS

T<sub>amb</sub> = -40°C to +85°C, V<sub>CC</sub> = 5V ±10%, V<sub>SS</sub> = 0V (Philips North America Parts Only)  
T<sub>amb</sub> = -40°C to +125°C, V<sub>CC</sub> = 5V ±10%, V<sub>SS</sub> = 0V (Philips Parts Only)

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			MIN	MAX	
V <sub>IL</sub>	Input low voltage, except EA (Philips North America)		-0.5	0.2V <sub>CC</sub> -0.15	V
V <sub>IL</sub>	Input low voltage, except EA (Philips)		-0.5	0.2V <sub>CC</sub> -0.25	V
V <sub>IL1</sub>	Input low voltage to EA		-0.5	0.2V <sub>CC</sub> -0.45	V
V <sub>IH</sub>	Input high voltage, except XTAL1, RST		0.2V <sub>CC</sub> +1	V <sub>CC</sub> +0.5	V
V <sub>IH1</sub>	Input high voltage to XTAL1, RST		0.7V <sub>CC</sub> +0.1	V <sub>CC</sub> +0.5	V
I <sub>IL</sub>	Logical 0 input current, ports 1, 2, 3	V <sub>IN</sub> = 0.45V		-75	μA
I <sub>TL</sub>	Logical 1-to-0 transition current, ports 1, 2, 3	V <sub>IN</sub> = 2.0V		-750	μA
I <sub>CC</sub>	Power supply current: Active mode <sup>1</sup> @ 16MHz (Philips) Active mode @ 12MHz (Philips North America) Idle mode <sup>2</sup> @ 16MHz (Philips) Idle mode @ 12MHz (Philips North America) Power-down mode <sup>3</sup> (Philips) Power-down mode (Philips North America)	V <sub>CC</sub> = 4.5-5.5V		25 20 6.5 5 75 50	mA mA mA mA μA μA

NOTES:

- The operating supply current is measured with all output pins disconnected; XTAL1 driven with t<sub>r</sub> = t<sub>f</sub> = 10ns; V<sub>IL</sub> = V<sub>SS</sub> + 0.5V; V<sub>IH</sub> = V<sub>CC</sub> - 0.5V; XTAL2 not connected; EA = RST = Port 0 = V<sub>CC</sub>.
- The idle mode supply current is measured with all output pins disconnected; XTAL1 driven with t<sub>r</sub> = t<sub>f</sub> = 10ns; V<sub>IL</sub> = V<sub>SS</sub> + 0.5V; V<sub>IH</sub> = V<sub>CC</sub> - 0.5V; XTAL2 not connected; EA = Port 0 = V<sub>CC</sub>; RST = V<sub>SS</sub>.
- The power-down current is measured with all output pins disconnected, XTAL2 not connected, EA = Port 0 = V<sub>CC</sub>; RST = V<sub>SS</sub>.

ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>

PARAMETER	RATING	UNIT
Operating temperature under bias	0 to +70 or -40 to +85	°C
Storage temperature range	-65 to +150	°C
Voltage on EA/V <sub>PP</sub> pin to V <sub>SS</sub>	0 to +13.0	V
Voltage on any other pin to V <sub>SS</sub>	-0.5 to +6.5	V
Maximum I <sub>OL</sub> per I/O pin	15	mA
Power dissipation (based on package heat transfer limitations, not device power consumption)	1.5	W

NOTES:

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
- This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
- Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V<sub>SS</sub> unless otherwise noted.

## CMOS single-chip 8-bit microcontroller 80C31/80C51/87C51

## DC ELECTRICAL CHARACTERISTICS

$T_{amb} = 0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$  or  $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 20\%$ ,  $V_{SS} = 0\text{V}$  (80C31/51) (12, 16, and 24MHz versions)

$T_{amb} = 0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$  or  $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$  (87C51) (80C31/80C51 33MHz version)

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYPICAL <sup>1</sup>	MAX	
$V_{IL}$	Input low voltage, except EA <sup>7</sup>		-0.5		$0.2V_{CC}-0.1$	V
$V_{IL1}$	Input low voltage to EA <sup>7</sup>		0		$0.2V_{CC}-0.3$	V
$V_{IH}$	Input high voltage, except XTAL1, RST <sup>7</sup>		$0.2V_{CC}+0.9$		$V_{CC}+0.5$	V
$V_{IH1}$	Input high voltage, XTAL1, RST <sup>7</sup>		$0.7V_{CC}$		$V_{CC}+0.5$	V
$V_{OL}$	Output low voltage, ports 1, 2, 3 <sup>11</sup>	$I_{OL} = 1.6\text{mA}^2$			0.45	V
$V_{OL1}$	Output low voltage, port 0, ALE, PSEN <sup>11</sup>	$I_{OL} = 3.2\text{mA}^2$			0.45	V
$V_{OH}$	Output high voltage, ports 1, 2, 3, ALE, PSEN <sup>3</sup>	$I_{OH} = -60\mu\text{A}$ , $I_{OH} = -25\mu\text{A}$ , $I_{OH} = -10\mu\text{A}$	2.4 $0.75V_{CC}$ $0.9V_{CC}$			V V V
$V_{OH1}$	Output high voltage (port 0 in external bus mode)	$I_{OH} = -800\mu\text{A}$ , $I_{OH} = -300\mu\text{A}$ , $I_{OH} = -80\mu\text{A}$	2.4 $0.75V_{CC}$ $0.9V_{CC}$			V V V
$I_{iL}$	Logical 0 input current, ports 1, 2, 3 <sup>7</sup>	$V_{IN} = 0.45\text{V}$			-50	$\mu\text{A}$
$I_{TL}$	Logical 1-to-0 transition current, ports 1, 2, 3 <sup>7</sup>	See note 4			-650	$\mu\text{A}$
$I_{LI}$	Input leakage current, port 0	$V_{IN} = V_{IL}$ or $V_{IH}$			$\pm 10$	$\mu\text{A}$
$I_{CC}$	Power supply current: <sup>7</sup> Active mode @ 12MHz <sup>8</sup> (Philips) Active mode @ 12MHz <sup>5</sup> (Philips North America) Idle mode @ 12MHz <sup>9</sup> (Philips) Idle mode @ 12MHz (Philips North America) Power-down mode <sup>10</sup> (Philips and Philips North America)	See note 6		11.5 1.3 3	18 19 4.4 4 50	$\text{mA}$ $\text{mA}$ $\text{mA}$ $\text{mA}$ $\mu\text{A}$
$R_{RST}$	Internal reset pull-down resistor (Philips North America) (Philips)		50 50		300 150	$\text{k}\Omega$ $\text{k}\Omega$
$C_{IO}$	Pin capacitance (12)				10	pF

## NOTES:

- Typical ratings are not guaranteed. The values listed are at room temperature, 5V.
- Capacitive loading on ports 0 and 2 may cause spurious noise to be superimposed on the  $V_{OL}$ s of ALE and ports 1 and 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100pF), the noise pulse on the ALE pin may exceed 0.8V. In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.  $I_{OL}$  can exceed these conditions provided that no single output sinks more than 5mA and no more than two outputs exceed the test conditions.
- Capacitive loading on ports 0 and 2 may cause the  $V_{OH}$  on ALE and PSEN to momentarily fall below the  $0.9V_{CC}$  specification when the address bits are stabilizing.
- Pins of ports 1, 2 and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when  $V_{IN}$  is approximately 2V.
- $I_{CCMAX}$  at other frequencies (for Philips North America parts) is given by: Active mode:  $I_{CCMAX} = 1.43 \times \text{FREQ} + 1.90$ ; Idle mode:  $I_{CCMAX} = 0.14 \times \text{FREQ} + 2.31$ , where FREQ is the external oscillator frequency in MHz.  $I_{CCMAX}$  is given in mA. See Figure 7.
- See Figures 8 through 11 for  $I_{CC}$  test conditions.
- For Philips North America parts when  $T_{amb} = -40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$  or Philips parts when  $T_{amb} = -40^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$ , see DC Electrical Characteristics table on previous page.
- The operating supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 10\text{ns}$ ;  $V_{IL} = V_{SS} + 0.5\text{V}$ ;  $V_{IH} = V_{CC} - 0.5\text{V}$ ; XTAL2 not connected; EA = RST = Port 0 =  $V_{CC}$ .
- The idle mode supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 10\text{ns}$ ;  $V_{IL} = V_{SS} + 0.5\text{V}$ ;  $V_{IH} = V_{CC} - 0.5\text{V}$ ; XTAL2 not connected; EA = Port 0 =  $V_{CC}$ ; RST =  $V_{SS}$ .
- The power-down current is measured with all output pins disconnected, XTAL2 not connected, EA = Port 0 =  $V_{CC}$ ; RST =  $V_{SS}$ .
- Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:  
Maximum  $I_{OL}$  per port pin: 15mA  
Maximum  $I_{OL}$  per 8-bit port: 26mA  
Maximum  $I_{OL}$  total for all outputs: 67mA  
If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
- Pin capacitance for the ceramic DIP package is 15pF maximum.

## CMOS single-chip 8-bit microcontroller 80C31/80C51/87C51

## AC ELECTRICAL CHARACTERISTICS FOR 12–33MHz PHILIPS NORTH AMERICA DEVICES

$T_{amb} = 0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$  or  $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 20\%$ ,  $V_{SS} = 0\text{V}$  (80C31/51)<sup>1, 2, 4</sup> (12, 16, and 24MHz versions)

$T_{amb} = 0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$  or  $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$  (87C51) (80C31/80C51 33MHz version)

SYMBOL	FIGURE	PARAMETER	VARIABLE CLOCK <sup>3</sup>		UNIT
			MIN	MAX	
$1/t_{CLCL}$		Oscillator frequency: <b>Speed Versions</b> SC80C31/51 C G P Y	3.5 3.5 3.5 3.5	12 16 24 33	MHz MHz MHz MHz
$t_{LHLL}$	1	ALE pulse width	$2t_{CLCL} - 40$		ns
$t_{AVLL}$	1	Address valid to ALE low	$t_{CLCL} - 13$		ns
$t_{LLAX}$	1	Address hold after ALE low	$t_{CLCL} - 20$		ns
$t_{LLIV}$	1	ALE low to valid instruction in		$4t_{CLCL} - 65$	ns
$t_{LLPL}$	1	ALE low to PSEN low	$t_{CLCL} - 13$		ns
$t_{PLPH}$	1	PSEN pulse width	$3t_{CLCL} - 20$		ns
$t_{PLIV}$	1	PSEN low to valid instruction in		$3t_{CLCL} - 45$	ns
$t_{PXIX}$	1	Input instruction hold after PSEN	0		ns
$t_{PXIZ}$	1	Input instruction float after PSEN		$t_{CLCL} - 10$	ns
$t_{AVIV}$	1	Address to valid instruction in		$5t_{CLCL} - 55$	ns
$t_{PLAZ}$	1	PSEN low to address float		10	ns
<b>Data Memory</b>					
$t_{RLRH}$	2, 3	RD pulse width	$6t_{CLCL} - 100$		ns
$t_{WLWH}$	2, 3	WR pulse width	$6t_{CLCL} - 100$		ns
$t_{RLDV}$	2, 3	RD low to valid data in		$5t_{CLCL} - 90$	ns
$t_{RHDX}$	2, 3	Data hold after RD	0		ns
$t_{RHDZ}$	2, 3	Data float after RD		$2t_{CLCL} - 28$	ns
$t_{LLDV}$	2, 3	ALE low to valid data in		$8t_{CLCL} - 150$	ns
$t_{AVDV}$	2, 3	Address to valid data in		$9t_{CLCL} - 165$	ns
$t_{LLWL}$	2, 3	ALE low to RD or WR low	$3t_{CLCL} - 50$	$3t_{CLCL} + 50$	ns
$t_{AVWL}$	2, 3	Address valid to WR low or RD low	$4t_{CLCL} - 75$		ns
$t_{QVWX}$	2, 3	Data valid to WR transition	$t_{CLCL} - 20$		ns
$t_{WHQX}$	2, 3	Data hold after WR	$t_{CLCL} - 20$		ns
$t_{RLAZ}$	2, 3	RD low to address float		0	ns
$t_{WHLH}$	2, 3	RD or WR high to ALE high	$t_{CLCL} - 20$	$t_{CLCL} + 25$	ns
<b>External Clock</b>					
$t_{CHCX}$	4	High time	12		ns
$t_{CLCX}$	4	Low time	12		ns
$t_{CLCH}$	4	Rise time		20	ns
$t_{CHCL}$	4	Fall time		20	ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- For all Philips North America speed versions only.
- Interfacing the 80C31/51 to devices with float times up to 50ns is permitted. This limited bus contention will not cause damage to port 0 drivers.



## CMOS single-chip 8-bit microcontroller 80C31/80C51/87C51

## AC ELECTRICAL CHARACTERISTICS FOR PHILIPS DEVICES

$T_{amb} = 0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 20\%$ ,  $V_{SS} = 0\text{V}$  (80C31/51)<sup>1, 2, 4, 5</sup>

SYMBOL	FIGURE	PARAMETER	VARIABLE CLOCK <sup>3</sup>		UNIT
			MIN	MAX	
$1/t_{CLCL}$		Oscillator frequency: <b>Speed Versions</b> PCB80C31/51 -2 PCA/PCB/PCF80C31/51 -3 PCB/PCF80C31/51 -4 PCB/FB80C31/51 -5	0.5 1.2 1.2 1.2	12 16 24 33	MHz MHz MHz MHz
$t_{LHLL}$	1	ALE pulse width	$2t_{CLCL}-40$		ns
$t_{AVLL}$	1	Address valid to ALE low	$t_{CLCL}-25$		ns
$t_{LLAX}$	1	Address hold after ALE low	$t_{CLCL}-25$		ns
$t_{LLIV}$	1	ALE low to valid instruction in		$4t_{CLCL}-65$	ns
$t_{LLPL}$	1	ALE low to PSEN low	$t_{CLCL}-25$		ns
$t_{PLPH}$	1	PSEN pulse width	$3t_{CLCL}-45$		ns
$t_{PLIV}$	1	PSEN low to valid instruction in		$3t_{CLCL}-60$	ns
$t_{PXIX}$	1	Input instruction hold after PSEN	0		ns
$t_{PXIZ}$	1	Input instruction float after PSEN		$t_{CLCL}-25$	ns
$t_{AVIV}$	1	Address to valid instruction in		$5t_{CLCL}-80$	ns
$t_{PLAZ}$	1	PSEN low to address float		10	ns
<b>Data Memory</b>					
$t_{RLRH}$	2, 3	RD pulse width	$6t_{CLCL}-100$		ns
$t_{WLWH}$	2, 3	WR pulse width	$6t_{CLCL}-100$		ns
$t_{RLDV}$	2, 3	RD low to valid data in		$5t_{CLCL}-90$	ns
$t_{RHDX}$	2, 3	Data hold after RD	0		ns
$t_{RHDZ}$	2, 3	Data float after RD		$2t_{CLCL}-28$	ns
$t_{LLDV}$	2, 3	ALE low to valid data in		$8t_{CLCL}-150$	ns
$t_{AVDV}$	2, 3	Address to valid data in		$9t_{CLCL}-165$	ns
$t_{LLWL}$	2, 3	ALE low to RD or WR low	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
$t_{AVWL}$	2, 3	Address valid to WR low or RD low	$4t_{CLCL}-75$		ns
$t_{QVWX}$	2, 3	Data valid to WR transition	$t_{CLCL}-30$		ns
$t_{WHQX}$	2, 3	Data hold after WR	$t_{CLCL}-25$		ns
$t_{RLAZ}$	2, 3	RD low to address float		0	ns
$t_{WHLH}$	2, 3	RD or WR high to ALE high	$t_{CLCL}-25$	$t_{CLCL}+25$	ns
<b>External Clock</b>					
$t_{CHCX}$	4	High time	15		ns
$t_{CLCX}$	4	Low time	15		ns
$t_{CLCH}$	4	Rise time		20	ns
$t_{CHCL}$	4	Fall time		20	ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- For all Philips speed versions only.
- Interfacing the 80C31/51 to devices with float times up to 30ns is permitted. This limited bus contention will not cause damage to port 0 drivers.
- $V_{CC} = 5\text{V} \pm 10\%$  for 33MHz.



## CMOS single-chip 8-bit microcontroller

## 80C31/80C51/87C51

## EXPLANATION OF THE AC SYMBOLS

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

A – Address

C – Clock

D – Input data

H – Logic level high

1 – Instruction (program memory contents)

L – Logic level low, or ALE

P - PSEN

Q – Output data

R –  $\overline{RD}$  signal

t - Time

V - Valid

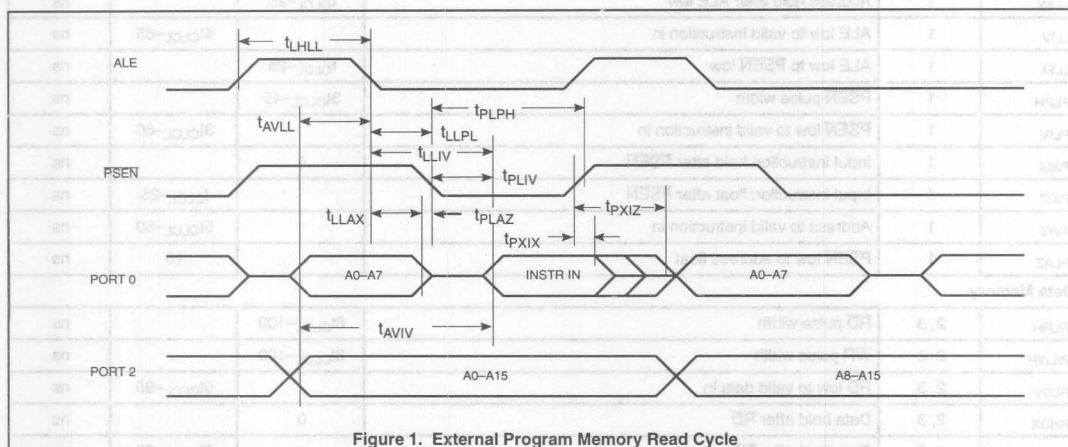
W-  $\overline{WR}$  signal

X – No longer a valid logic level

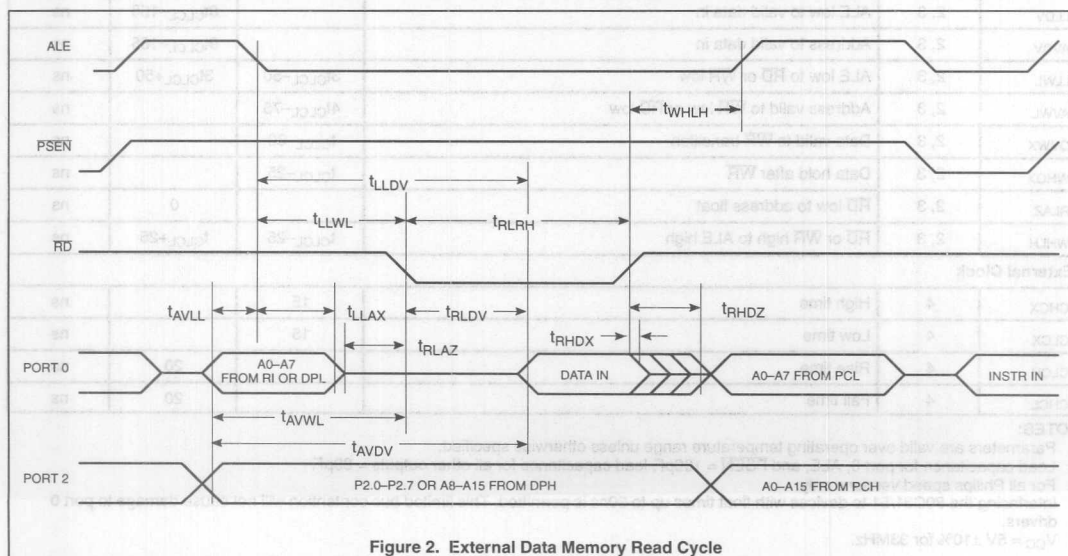
Z - Float

**Examples:**  $t_{AVLL}$  = Time for address valid to ALE low.

$t_{LLPL}$  =  
Time for ALE low to  
PSEN low



**Figure 1. External Program Memory Read Cycle**



**Figure 2. External Data Memory Read Cycle**

## CMOS single-chip 8-bit microcontroller 80C31/80C51/87C51

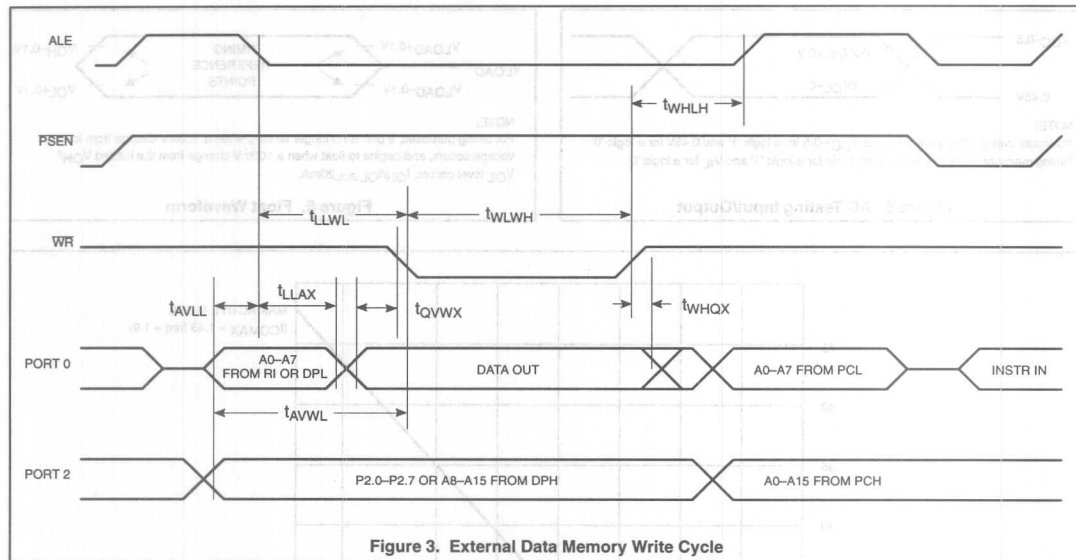


Figure 3. External Data Memory Write Cycle

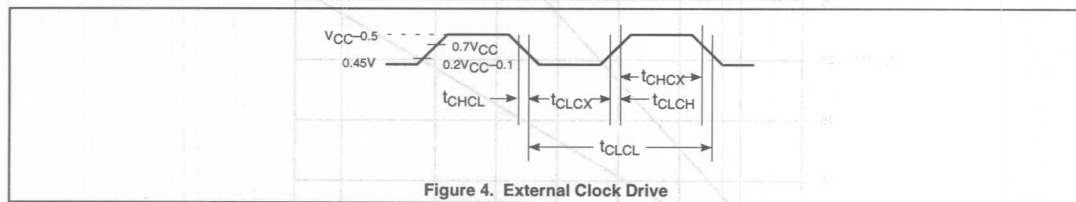
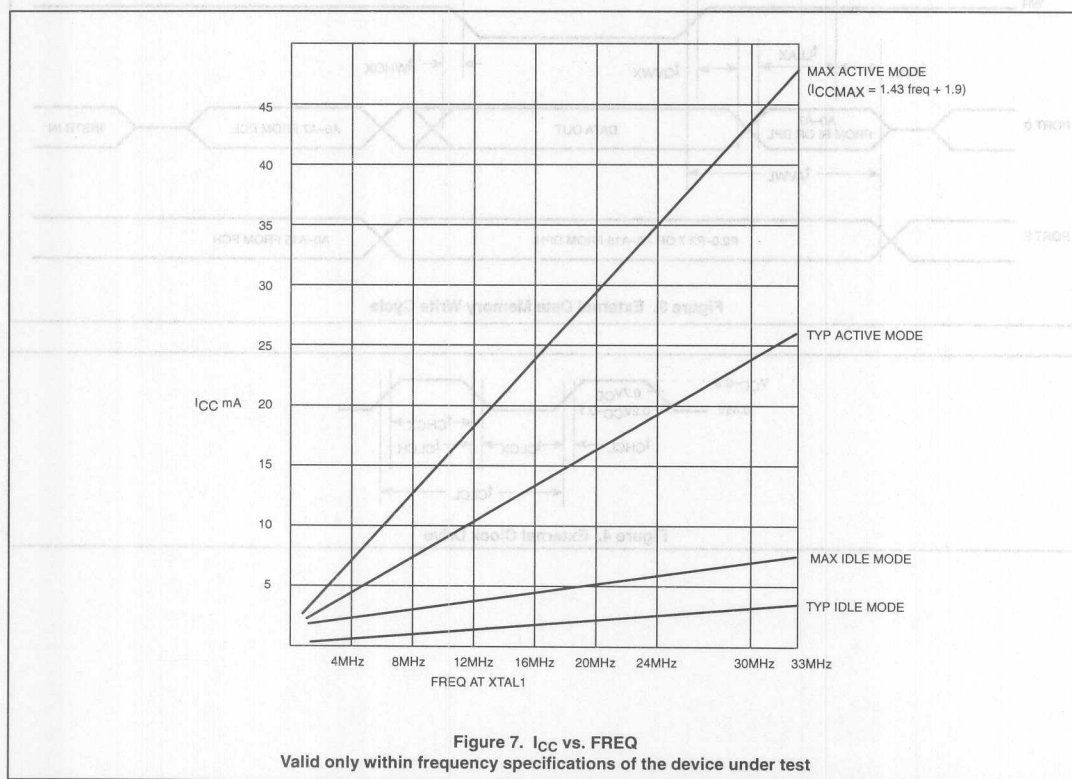
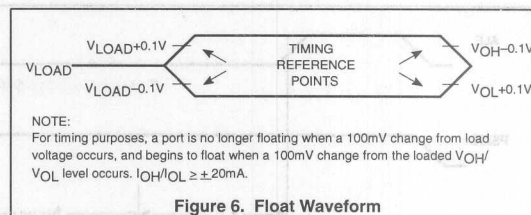
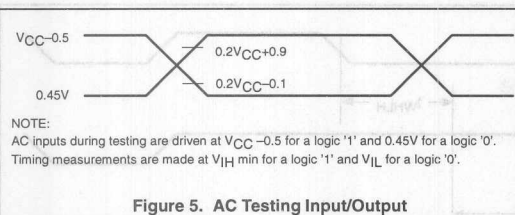
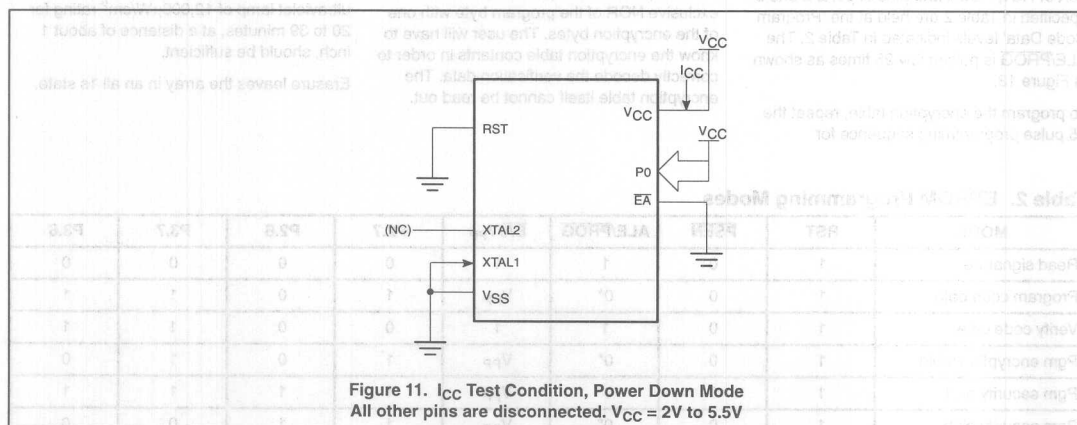
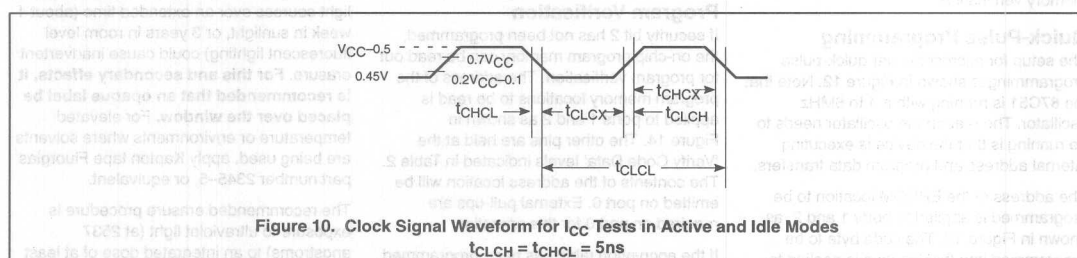
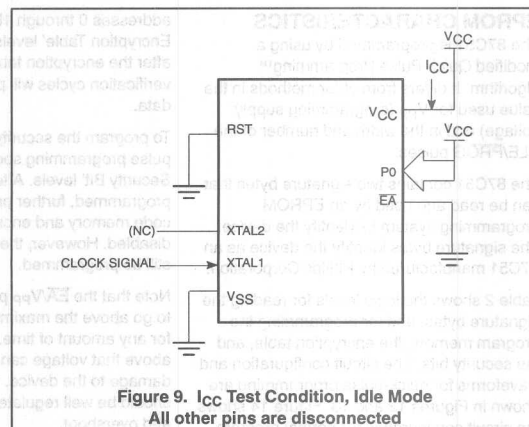
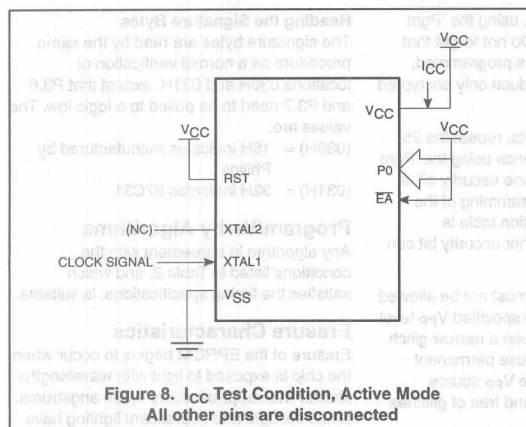


Figure 4. External Clock Drive



## CMOS single-chip 8-bit microcontroller

## 80C31/80C51/87C51



## CMOS single-chip 8-bit microcontroller 80C31/80C51/87C51

### EPROM CHARACTERISTICS

The 87C51 is programmed by using a modified Quick-Pulse Programming™ algorithm. It differs from older methods in the value used for  $V_{PP}$  (programming supply voltage) and in the width and number of the ALE/PROG pulses.

The 87C51 contains two signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes identify the device as an 87C51 manufactured by Philips Corporation.

Table 2 shows the logic levels for reading the signature bytes, and for programming the program memory, the encryption table, and the security bits. The circuit configuration and waveforms for quick-pulse programming are shown in Figures 12 and 13. Figure 14 shows the circuit configuration for normal program memory verification.

### Quick-Pulse Programming

The setup for microcontroller quick-pulse programming is shown in Figure 12. Note that the 87C51 is running with a 4 to 6MHz oscillator. The reason the oscillator needs to be running is that the device is executing internal address and program data transfers.

The address of the EPROM location to be programmed is applied to ports 1 and 2, as shown in Figure 12. The code byte to be programmed into that location is applied to port 0. RST, PSEN and pins of ports 2 and 3 specified in Table 2 are held at the 'Program Code Data' levels indicated in Table 2. The ALE/PROG is pulsed low 25 times as shown in Figure 13.

To program the encryption table, repeat the 25 pulse programming sequence for

addresses 0 through 1FH, using the 'Pgm Encryption Table' levels. Do not forget that after the encryption table is programmed, verification cycles will produce only encrypted data.

To program the security bits, repeat the 25 pulse programming sequence using the 'Pgm Security Bit' levels. After one security bit is programmed, further programming of the code memory and encryption table is disabled. However, the other security bit can still be programmed.

Note that the  $\overline{EA}/V_{PP}$  pin must not be allowed to go above the maximum specified  $V_{PP}$  level for any amount of time. Even a narrow glitch above that voltage can cause permanent damage to the device. The  $V_{PP}$  source should be well regulated and free of glitches and overshoot.

### Program Verification

If security bit 2 has not been programmed, the on-chip program memory can be read out for program verification. The address of the program memory locations to be read is applied to ports 1 and 2 as shown in Figure 14. The other pins are held at the 'Verify Code Data' levels indicated in Table 2. The contents of the address location will be emitted on port 0. External pull-ups are required on port 0 for this operation.

If the encryption table has been programmed, the data presented at port 0 will be the exclusive NOR of the program byte with one of the encryption bytes. The user will have to know the encryption table contents in order to correctly decode the verification data. The encryption table itself cannot be read out.

### Reading the Signature Bytes

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. The values are:

(030H) = 15H indicates manufactured by Philips

(031H) = 92H indicates 87C51

### Program/Verify Algorithms

Any algorithm in agreement with the conditions listed in Table 2, and which satisfies the timing specifications, is suitable.

### Erase Characteristics

Erase of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent erasure. **For this and secondary effects, it is recommended that an opaque label be placed over the window.** For elevated temperature or environments where solvents are being used, apply Kapton tape Fluorglas part number 2345-5, or equivalent.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 angstroms) to an integrated dose of at least 15W-sec/cm<sup>2</sup>. Exposing the EPROM to an ultraviolet lamp of 12,000µW/cm<sup>2</sup> rating for 20 to 39 minutes, at a distance of about 1 inch, should be sufficient.

Erase leaves the array in an all 1s state.

Table 2. EPROM Programming Modes

MODE	RST	PSEN	ALE/PROG	$\overline{EA}/V_{PP}$	P2.7	P2.6	P3.7	P3.6
Read signature	1	0	1	1	0	0	0	0
Program code data	1	0	0*	$V_{PP}$	1	0	1	1
Verify code data	1	0	1	1	0	0	1	1
Pgm encryption table	1	0	0*	$V_{PP}$	1	0	1	0
Pgm security bit 1	1	0	0*	$V_{PP}$	1	1	1	1
Pgm security bit 2	1	0	0*	$V_{PP}$	1	1	0	0

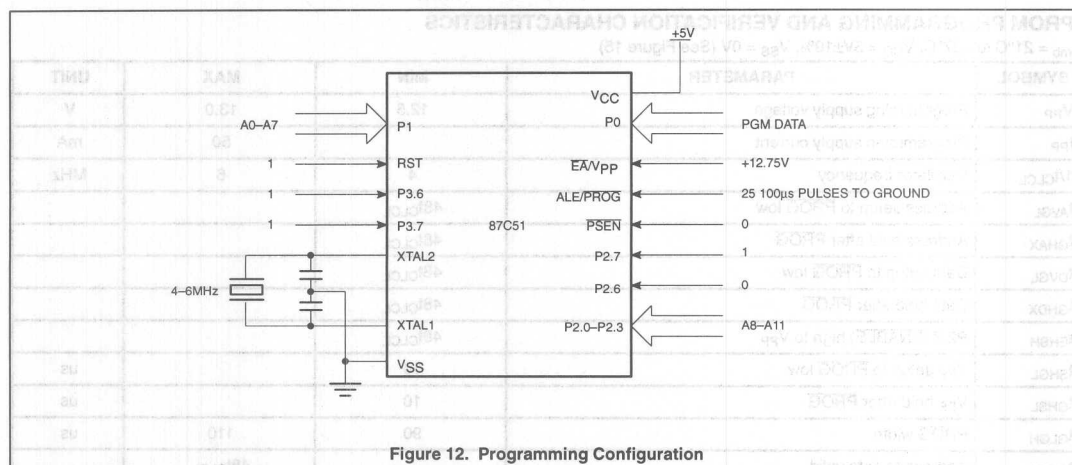
#### NOTES:

- '0' = Valid low for that pin, '1' = valid high for that pin.
- $V_{PP}$  = 12.75V  $\pm$  0.25V.
- $V_{CC}$  = 5V  $\pm$  10% during programming and verification.
- \*ALE/PROG receives 25 programming pulses while  $V_{PP}$  is held at 12.75V. Each programming pulse is low for 100µs ( $\pm$ 10µs) and high for a minimum of 10µs.

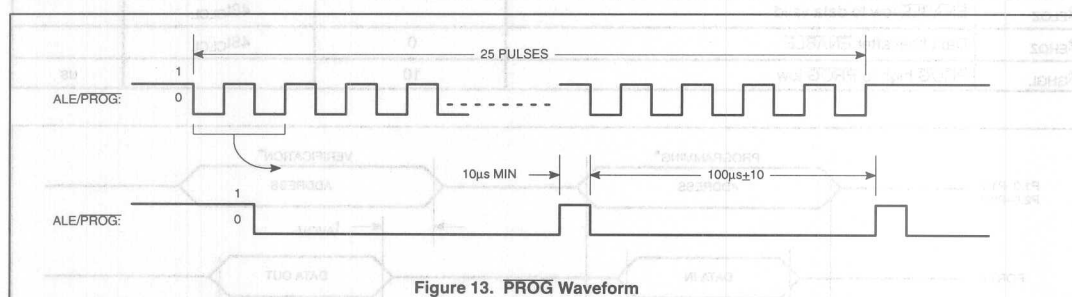
™Trademark phrase of Intel Corporation.

## CMOS single-chip 8-bit microcontroller

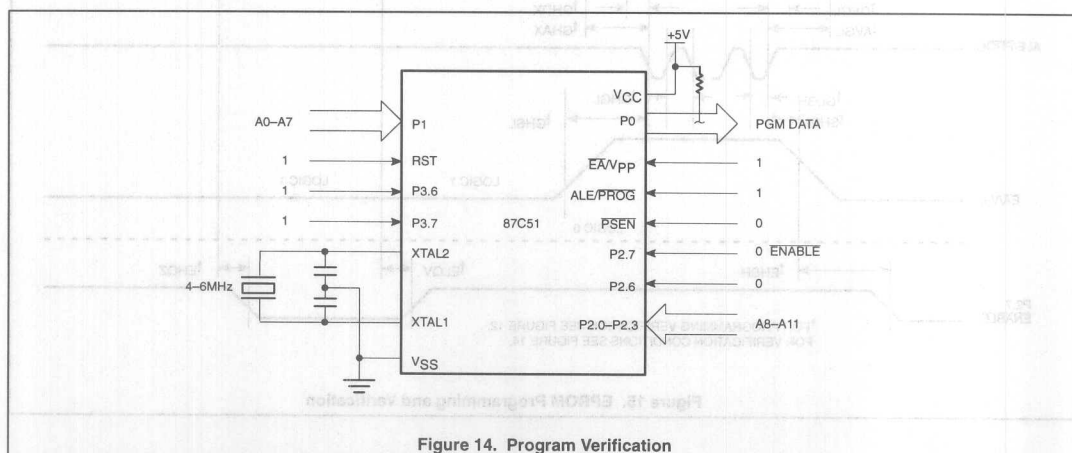
80C31/80C51/87C51



**Figure 12. Programming Configuration**



**Figure 13.  $\overline{\text{PROG}}$  Waveform**



**Figure 14. Program Verification**



## CMOS single-chip 8-bit microcontroller 80C31/80C51/87C51

## EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

$T_{amb} = 21^{\circ}\text{C}$  to  $+27^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$  (See Figure 15)

SYMBOL	PARAMETER	MIN	MAX	UNIT
$V_{PP}$	Programming supply voltage	12.5	13.0	V
$I_{PP}$	Programming supply current		50	mA
$1/t_{CLCL}$	Oscillator frequency	4	6	MHz
$t_{AVGL}$	Address setup to PROG low	$48t_{CLCL}$		
$t_{GHAX}$	Address hold after PROG	$48t_{CLCL}$		
$t_{DVGL}$	Data setup to PROG low	$48t_{CLCL}$		
$t_{GHDx}$	Data hold after PROG	$48t_{CLCL}$		
$t_{EHS}$	P2.7 (ENABLE) high to $V_{PP}$	$48t_{CLCL}$		
$t_{SHGL}$	$V_{PP}$ setup to PROG low	10		us
$t_{GHSL}$	$V_{PP}$ hold after PROG	10		us
$t_{GLGH}$	PROG width	90	110	us
$t_{AVQV}$	Address to data valid		$48t_{CLCL}$	
$t_{ELOZ}$	ENABLE low to data valid		$48t_{CLCL}$	
$t_{EHQZ}$	Data float after ENABLE	0	$48t_{CLCL}$	
$t_{GHGL}$	PROG high to PROG low	10		us

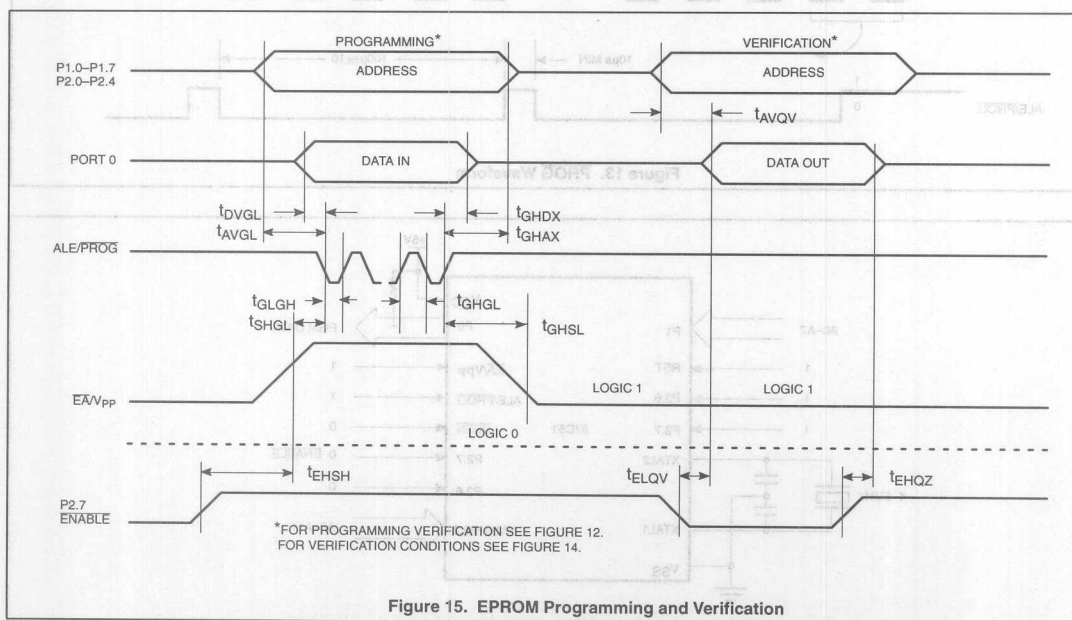


Figure 15. EPROM Programming and Verification

# CMOS single-chip 8-bit microcontroller 83C51FA/87C51FA

## DESCRIPTION

The 83C51FA and 87C51FA (hereafter generically referred to as 8XC51FA) Single-Chip 8-Bit Microcontrollers are manufactured in an advanced CMOS process and are derivatives of the 80C51 microcontroller family. The 8XC51FA has the same instruction set as the 80C51.

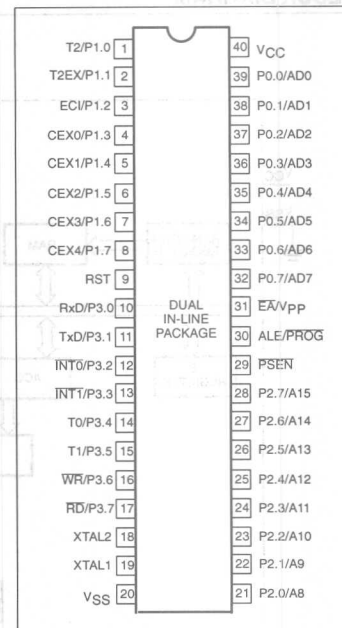
This device provides architectural enhancements that make it applicable in a variety of applications for general control systems. The 8XC51FA contains 8k × 8 memory, a volatile 256 × 8 read/write data memory, four 8-bit I/O ports, three 16-bit timer/event counters, a Programmable Counter Array (PCA), a multi-source, two-priority-level, nested interrupt structure, an enhanced UART and on-chip oscillator and timing circuits. For systems that require extra capability, the 8XC51FA can be expanded using standard TTL compatible memories and logic.

Its added features make it an even more powerful microcontroller for applications that require pulse width modulation, high-speed I/O and up/down counting capabilities such as motor control. It also has a more versatile serial channel that facilitates multiprocessor communications.

## FEATURES

- 80C51 central processing unit
- 8k × 8 EPROM (87C51FA)  
8k × 8 ROM (83C51FA)
  - expandable externally to 64k bytes
  - Quick Pulse programming algorithm
  - Two level program security system
- 256 × 8 RAM, expandable externally to 64k bytes
- Three 16-bit timer/counters
  - T2 is an up/down counter
- Programmable Counter Array (PCA)
  - High speed output
  - Capture/compare
  - Pulse Width Modulator
  - Watchdog Timer
- Four 8-bit I/O ports
- Full-duplex enhanced UART
  - Framing error detection
  - Automatic address recognition
- Power control modes
  - Idle mode
  - Power-down mode
- Once (On Circuit Emulation) Mode
- Five package styles
- OTP package available

## PIN CONFIGURATIONS



## ORDERING INFORMATION

ROM	EPROM		TEMPERATURE RANGE °C AND PACKAGE <sup>1</sup>	FREQ. (MHz)	DRAWING NUMBER
S83C51FA-4N40	S87C51FA-4N40	OTP	0 to +70, 40-Pin Plastic Dual In-line Package	3.5 to 16	0415C
	S87C51FA-4F40	UV	0 to +70, 40-Pin Ceramic Dual In-line Package w/Window	3.5 to 16	0590B
S83C51FA-4A44	S87C51FA-4A44	OTP	0 to +70, 44-Pin Plastic Leaded Chip Carrier	3.5 to 16	0403G
	S87C51FA-4K44	UV	0 to +70, 44-Pin Ceramic Leaded Chip Carrier w/Window	3.5 to 16	1472A
S83C51FA-4B44	S87C51FA-4B44	OTP	0 to +70, 44-Pin Plastic Quad Flat Pack	3.5 to 16	1118D
S83C51FA-5N40	S87C51FA-5N40	OTP	-40 to +85, 40-Pin Plastic Dual In-line Package	3.5 to 16	0415C
	S87C51FA-5F40	UV	-40 to +85, 40-Pin Ceramic Dual In-line Package w/Window	3.5 to 16	0590B
S83C51FA-5A44	S87C51FA-5A44	OTP	-40 to +85, 44-Pin Plastic Leaded Chip Carrier	3.5 to 16	0403G
	S87C51FA-5K44	UV	-40 to +85, 44-Pin Ceramic Leaded Chip Carrier w/Window	3.5 to 16	1472A
S83C51FA-5B44	S87C51FA-5B44	OTP	-40 to +85, 44-Pin Plastic Quad Flat Pack	3.5 to 16	1118D
S83C51FA-AN40	S87C51FA-AN40	OTP	0 to +70, 40-Pin Plastic Dual In-line Package	3.5 to 24	0415C
	S87C51FA-AF40	UV	0 to +70, 40-Pin Ceramic Dual In-line Package w/Window	3.5 to 24	0590B
S83C51FA-AA44	S87C51FA-AA44	OTP	0 to +70, 44-Pin Plastic Leaded Chip Carrier	3.5 to 24	0403G
	S87C51FA-AK44	UV	0 to +70, 44-Pin Ceramic Leaded Chip Carrier w/Window	3.5 to 24	1472A
S83C51FA-BN40	S87C51FA-BN40	OTP	-40 to +85, 40-Pin Plastic Dual In-line Package	3.5 to 24	0415C
	S87C51FA-BF40	UV	-40 to +85, 40-Pin Ceramic Dual In-line Package w/Window	3.5 to 24	0590B
S83C51FA-BA44	S87C51FA-BA44	OTP	-40 to +85, 44-Pin Plastic Leaded Chip Carrier	3.5 to 24	0403G
	S87C51FA-BK44	UV	-40 to +85, 44-Pin Ceramic Leaded Chip Carrier w/Window	3.5 to 24	1472A

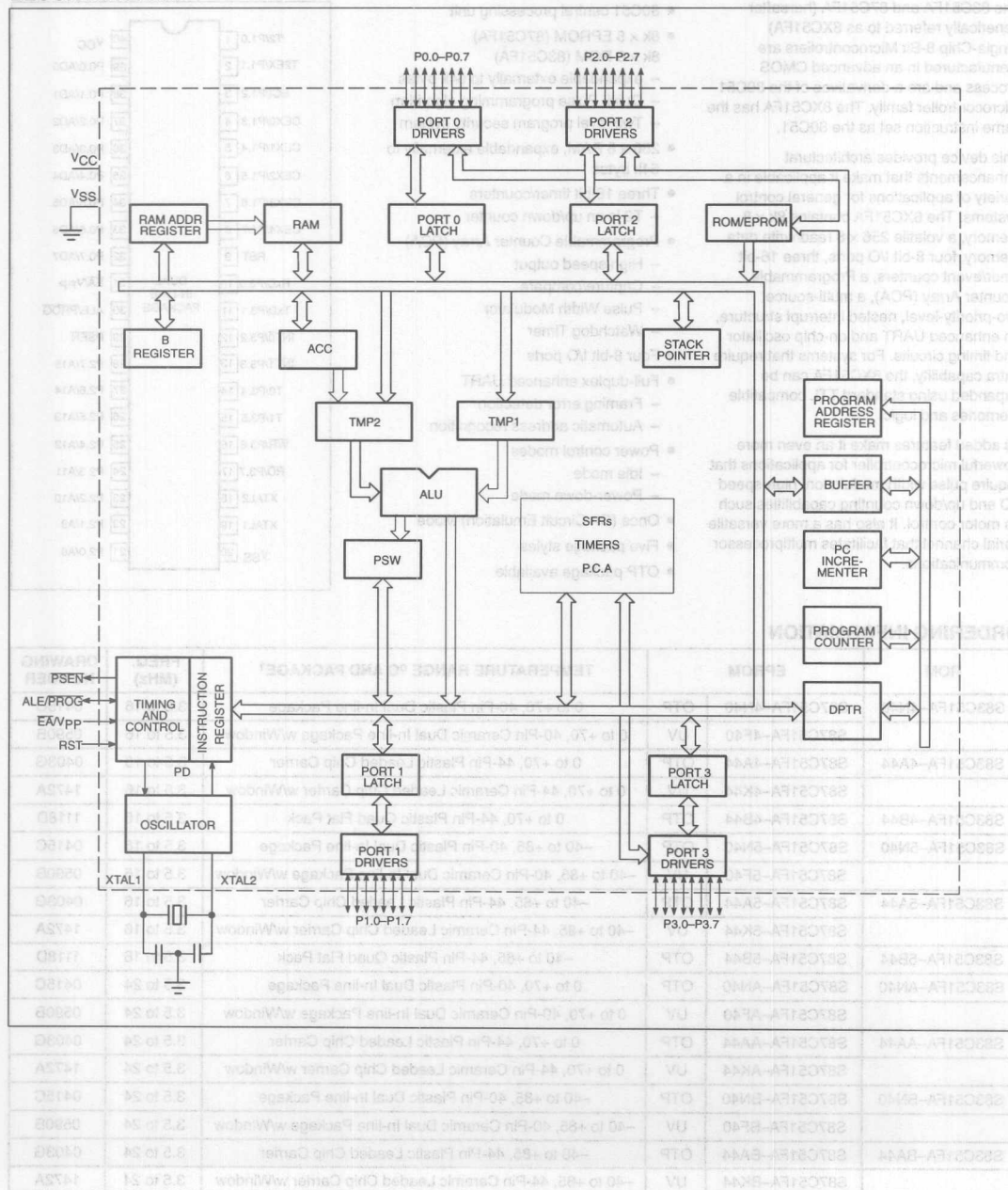
### NOTE:

1. OTP = One Time Programmable EPROM. UV = Erasable EPROM.

## CMOS single-chip 8-bit microcontroller

83C51FA/87C51FA

## BLOCK DIAGRAM

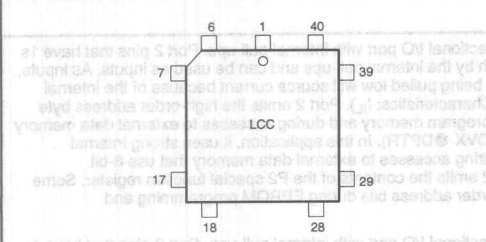


## CMOS single-chip 8-bit microcontroller

83C51FA/87C51FA

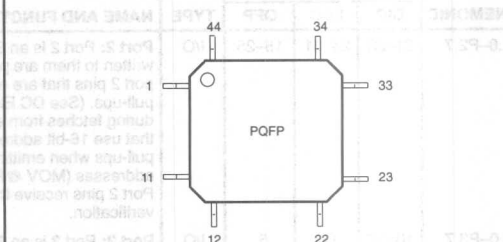
## CERAMIC AND PLASTIC LEADED CHIP CARRIER

## PIN FUNCTIONS



Pin	Function	Pin	Function
1	NC	23	NC
2	P1.0/T2	24	P2.0/A8
3	P1.1/T2EX	25	P2.1/A9
4	P1.2/ECI	26	P2.2/A10
5	P1.3/CEX0	27	P2.3/A11
6	P1.4/CEX1	28	P2.4/A12
7	P1.5/CEX2	29	P2.5/A13
8	P1.6/CEX3	30	P2.6/A14
9	P1.7/CEX4	31	P2.7/A15
10	RST	32	PSEN
11	P3.0/RxD	33	ALE/PROG
12	NC	34	NC
13	P3.1/TxD	35	EA/Vpp
14	P3.2/INT0	36	P0.7/AD7
15	P3.3/INT1	37	P0.6/AD6
16	P3.4/T0	38	P0.5/AD5
17	P3.5/T1	39	P0.4/AD4
18	P3.6/WR	40	P0.3/AD3
19	P3.7/RD	41	P0.2/AD2
20	XTAL2	42	P0.1/AD1
21	XTAL1	43	P0.0/AD0
22	VSS	44	VCC

## PLASTIC QUAD FLAT PACK PIN FUNCTIONS



Pin	Function	Pin	Function
1	P1.5/CEX2	23	P2.5/A13
2	P1.6/CEX3	24	P2.6/A14
3	P1.7/CEX4	25	P2.7/A15
4	RST	26	PSEN
5	P3.0/RxD	27	ALE/PROG
6	NC	28	NC
7	P3.1/TxD	29	EA/Vpp
8	P3.2/INT0	30	P0.7/AD7
9	P3.3/INT1	31	P0.6/AD6
10	P3.4/T0	32	P0.5/AD5
11	P3.5/T1	33	P0.4/AD4
12	P3.6/WR	34	P0.3/AD3
13	P3.7/RD	35	P0.2/AD2
14	XTAL2	36	P0.1/AD1
15	XTAL1	37	P0.0/AD0
16	VSS	38	VCC
17	NC	39	NC
18	P2.0/A8	40	P1.0/T2
19	P2.1/A9	41	P1.1/T2EX
20	P2.2/A10	42	P1.2/ECI
21	P2.3/A11	43	P1.3/CEX0
22	P2.4/A12	44	P1.4/CEX1

## PIN DESCRIPTIONS

MNEMONIC	PIN NUMBER			TYPE	NAME AND FUNCTION
	DIP	LCC	QFP		
Vss	20	22	16	1	Ground: 0V reference.
Vcc	40	44	38	1	<b>Power Supply:</b> This is the power supply voltage for normal, idle, and power-down operation.
P0.0-0.7	39-32	43-36	37-30	I/O	<b>Port 0:</b> Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s. Port 0 also outputs the code bytes during program verification and receives code bytes during EPROM programming. External pull-ups are required during program verification.
P1.0-P1.7	1-8	2-9	40-44, 1-3	I/O	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pull-ups, except P1.6 and P1.7 which are open drain. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 1 also receives the low-order address byte during program memory verification. Alternate functions include: <b>T2 (P1.0):</b> Timer/Counter 2 external count input/Clockout <b>T2EX (P1.1):</b> Timer/Counter 2 Reload/Capture/Direction Control <b>ECI (P1.2):</b> External Clock Input to the PCA <b>CEX0 (P1.3):</b> Capture/Compare External I/O for PCA module 0 <b>CEX1 (P1.4):</b> Capture/Compare External I/O for PCA module 1 <b>CEX2 (P1.5):</b> Capture/Compare External I/O for PCA module 2 <b>CEX3 (P1.6):</b> Capture/Compare External I/O for PCA module 3 <b>CEX4 (P1.7):</b> Capture/Compare External I/O for PCA module 4
	1	2	40	1	
	2	3	41	1	
	3	4	42	1	
	4	5	43	I/O	
	5	6	44	I/O	
	6	7	1	I/O	
	7	8	2	I/O	
	8	9	3	I/O	

## CMOS single-chip 8-bit microcontroller

83C51FA/87C51FA

## PIN DESCRIPTIONS (Continued)

MNEMONIC	PIN NUMBER			TYPE	NAME AND FUNCTION
	DIP	LCC	QFP		
P2.0–P2.7	21–28	24–31	18–25	I/O	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: $I_{IL}$ ). Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register. Some Port 2 pins receive the high order address bits during EPROM programming and verification.
P3.0–P3.7	10–17	11, 13–19	5, 7–13	I/O	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: $I_{IL}$ ). Port 3 also serves the special features of the 80C51 family, as listed below: <b>RxD (P3.0):</b> Serial input port <b>TxD (P3.1):</b> Serial output port <b>INT0 (P3.2):</b> External interrupt <b>INT1 (P3.3):</b> External interrupt <b>T0 (P3.4):</b> Timer 0 external input <b>T1 (P3.5):</b> Timer 1 external input <b>WR (P3.6):</b> External data memory write strobe <b>RD (P3.7):</b> External data memory read strobe
RST	9	10	4	I	<b>Reset:</b> A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to $V_{SS}$ permits a power-on reset using only an external capacitor to $V_{CC}$ .
ALE/PROG	30	33	27	I/O	<b>Address Latch Enable/Program Pulse:</b> Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory. This pin is also the program pulse input (PROG) during EPROM programming.
PSEN	29	32	26	O	<b>Program Store Enable:</b> The read strobe to external program memory. When the 8XC51FA is executing code from the external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory. PSEN is not activated during fetches from internal program memory.
EA/V <sub>PP</sub>	31	35	29	I	<b>External Access Enable/Programming Supply Voltage:</b> EA must be externally held low to enable the device to fetch code from external program memory locations 0000H to 1FFFFH. If EA is held high, the device executes from internal program memory unless the program counter contains an address greater than 1FFFFH. This pin also receives the 12.75V programming supply voltage ( $V_{PP}$ ) during EPROM programming. If security bit 1 is programmed, EA will be internally latched on Reset.
XTAL1	19	21	15	I	<b>Crystal 1:</b> Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	20	14	O	<b>Crystal 2:</b> Output from the inverting oscillator amplifier.

## NOTE:

To avoid "latch-up" effect at power-on, the voltage on any pin at any time must not be higher than  $V_{CC} + 0.5V$  or  $V_{SS} - 0.5V$ , respectively.

## TIMER 2

This is a 16-bit up or down counter, which can be operated as either a timer or event counter. It can be operated in one of three different modes (autoreload, capture or as the baud rate generator for the UART).

In the autoreload mode the Timer can be set to count up or down by setting or clearing the bit DCEN in the T2CON Special Function Register. The SFR's RCAP2H and RCAP2L are used to reload the Timer upon overflow or a 1-to-0 transition on the T2EX input (P1.1).

In the Capture mode Timer 2 can either set TF2 and generate an interrupt or capture its value. To capture Timer 2 in response to a 1-to-0 transition on the T2EX input, the EXEN2 bit in the T2CON must be set. Timer 2 is then captured in SFR's RCAP2H and RCAP2L.

As the baud rate generator, Timer 2 is selected by setting TCLK and/or RCLK in T2CON. As the baud rate generator Timer 2 is incremented at  $1/2$  the oscillator frequency.

## ENHANCED UART

The 8XC51FA UART has all of the capabilities of the standard 80C51 UART plus Framing Error Detection and Automatic Address Recognition. As in the 80C51, all four modes of operation are supported as well as the 9th bit in modes 2 and 3 that can be used to facilitate multiprocessor communication.

The Framing Error Detection allows the UART to look for missing stop bits. If a Stop



## CMOS single-chip 8-bit microcontroller

## 83C51FA/87C51FA

bit is missing, the FE bit in the SCON SFR is set. The FE bit can be checked after each transmission to detect communication errors. The FE bit can only be cleared by software and is not affected by a valid stop bit.

Automatic Address Recognition is used to reduce the CPU service time for the serial port. The CPU only needs to service the UART when it is addressed and, with this done by the on-chip circuitry, the need for software overhead is greatly reduced. This mode works similar to the 9th bit communication mode, except that uses only 8 bits and the Stop bit is used to cause the RI bit to be set. There are two SFRs associated with this mode. They are SADDR, which holds the slave address and SADEN, which contains a mask that allows selective masking of the slave address so that broadcast addresses can be used.

### PROGRAMMABLE COUNTER ARRAY

The PCA is a sophisticated free-running 16 bit Timer/Counter that drives 5 modules that can be individually configured as Capture inputs, software timers, high speed outputs, or pulse width modulated outputs. In addition, module 4 can be configured as a software controlled watchdog timer.

The Timer portion of the PCA can be configured to run in one of four different modes. The modes are:  $1/2$  the oscillator frequency,  $1/4$  the oscillator frequency, Timer 0 overflows, or from the ECI input.

For the Capture/Compare mode each of the modules has a pair of registers associated with it called CCAPnH and CCAPnL (where  $n = 0, 1, 2, 3, 4$  depending on the module). Both positive and negative transitions can be captured. This means that the PCA has the flexibility to measure phase differences, duty cycles, pulse widths and a wide variety of other digital pulse characteristics.

In the 16-bit software timer mode each of the modules can generate an interrupt upon a compare.

For applications that require accurate pulse widths and edges the PCA modules can be used as High Speed Outputs (HSO). The PCA toggles the appropriate CEXn pin when there is a match between the PCA timer and the modules compare registers.

The pulse width modulator mode for the PCA allows the conversion of digital information into analog signals. Each of the 5 modules can be used in this mode. The frequency of the PWM depends on the clock source for the PCA. The 8-bit PWM output is generated by comparing the low byte of the PCA (CL)

with the module's CCAPnL SFR. When  $CL < CCAPnL$ , the output is high. When  $CL > CCAPnL$ , the output is low.

### POWER OFF FLAG

The Power Off Flag (POF) is set by on-chip circuitry when the  $V_{CC}$  level on the 8XC51FA rises from 0 to 5V. The POF bit can be set or cleared by software allowing a user to determine if the reset is the result of a power-on or a warm start after powerdown. The  $V_{CC}$  level must remain above 3V for the POF to remain unaffected by the  $V_{CC}$  level.

### OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

### Reset

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. To insure a good power-on reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few milliseconds) plus two machine cycles. At power-on, the voltage on  $V_{CC}$  and RST must come up at the same time for a proper start-up.

### Idle Mode

In the idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

### Power-Down Mode

To save even more power, a Power Down mode can be invoked by software. In this

mode, the oscillator is stopped and the instruction that invoked Power Down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power Down mode is terminated.

On the 8XC51FA either a hardware reset or external interrupt can use an exit from Power Down. Reset redefines all the SFRs but does not change the on-chip RAM. An external interrupt allows both the SFRs and the on-chip RAM to retain their values.

To properly terminate Power Down the reset or external interrupt should not be executed before  $V_{CC}$  is restored to its normal operating level and must be held active long enough for the oscillator to restart and stabilize (normally less than 10ms).

With an external interrupt, INT0 and INT1 must be enabled and configured as level-sensitive. Holding the pin low restarts the oscillator but bringing the pin back high completes the exit. Once the interrupt is serviced, the next instruction to be executed after RETI will be the one following the instruction that put the device into Power Down.

### Design Consideration

- When the idle mode is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

### ONCE™ Mode

The ONCE ("On-Circuit Emulation") Mode facilitates testing and debugging of systems using the 8XC51FA without the 8XC51FA having to be removed from the circuit. The ONCE Mode is invoked by:

1. Pull ALE low while the device is in reset and PSEN is high;
2. Hold ALE low as RST is deactivated.

While the device is in ONCE Mode, the Port 0 pins go into a float state, and the other port pins and ALE and PSEN are weakly pulled high. The oscillator circuit remains active. While the 8XC51FA is in this mode, an emulator or test CPU can be used to drive the circuit. Normal operation is restored when a normal reset is applied.



## CMOS single-chip 8-bit microcontroller

## 83C51FA/87C51FA

Table 1. External Pin Status During Idle and Power-Down Mode

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>

PARAMETER	RATING	UNIT
Operating temperature under bias	0 to +70 or -40 to +85	°C
Storage temperature range	-65 to +150	°C
Voltage on EA/V <sub>PP</sub> pin to V <sub>SS</sub>	0 to +13.0	V
Voltage on any other pin to V <sub>SS</sub>	-0.5 to +6.5	V
Maximum I <sub>OL</sub> per I/O pin	15	mA
Power dissipation (based on package heat transfer limitations, not device power consumption)	1.5	W

## NOTES:

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
- This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
- Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V<sub>SS</sub> unless otherwise noted.

## Electrical Deviations from Commercial Specifications for Extended Temperature Range

DC and AC parameters not included here are the same as in the commercial temperature range table.

## DC ELECTRICAL CHARACTERISTICS

T<sub>amb</sub> = -40°C to +85°C, V<sub>CC</sub> = 5V ±10%, V<sub>SS</sub> = 0V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			MIN	MAX	
V <sub>IL</sub>	Input low voltage, except EA		-0.5	0.2V <sub>CC</sub> -0.15	V
V <sub>IL1</sub>	Input low voltage to EA		0	0.2V <sub>CC</sub> -0.35	V
V <sub>IH</sub>	Input high voltage, except XTAL1, RST		0.2V <sub>CC</sub> +1	V <sub>CC</sub> +0.5	V
V <sub>IH1</sub>	Input high voltage to XTAL1, RST		0.7V <sub>CC</sub> +0.1	V <sub>CC</sub> +0.5	V
I <sub>IL</sub>	Logical 0 input current, ports 1, 2, 3	V <sub>IN</sub> = 0.45V		-75	µA
I <sub>TL</sub>	Logical 1-to-0 transition current, ports 1, 2, 3	V <sub>IN</sub> = 2.0V		-750	µA
I <sub>CC</sub>	Power supply current: Active mode Idle mode Power-down mode	V <sub>CC</sub> = 4.5-5.5V, Frequency range = 3.5 to 16MHz		19 6 50	mA mA µA

## CMOS single-chip 8-bit microcontroller

## 83C51FA/87C51FA

## DC ELECTRICAL CHARACTERISTICS

$T_{amb} = 0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$  or  $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP <sup>1</sup>	MAX	
$V_{IL}$	Input low voltage, except EA <sup>7</sup>		-0.5		$0.2V_{CC}-0.1$	V
$V_{IL1}$	Input low voltage to EA <sup>7</sup>		0		$0.2V_{CC}-0.3$	V
$V_{IH}$	Input high voltage, except XTAL1, RST <sup>7</sup>		$0.2V_{CC}+0.9$		$V_{CC}+0.5$	V
$V_{IH1}$	Input high voltage, XTAL1, RST <sup>7</sup>		$0.7V_{CC}$		$V_{CC}+0.5$	V
$V_{OL}$	Output low voltage, ports 1, 2, 3 <sup>9</sup>	$I_{OL} = 100\mu\text{A}$ $I_{OL} = 1.6\text{mA}^2$ $I_{OL} = 3.5\text{mA}$			0.3 0.45 1.0	V V V
$V_{OL1}$	Output low voltage, port 0, ALE, PSEN <sup>9</sup>	$I_{OL} = 200\mu\text{A}$ $I_{OL} = 3.2\text{mA}^2$ $I_{OL} = 7.0\text{mA}$			0.3 0.45 1.0	V V V
$V_{OH}$	Output high voltage, ports 1, 2, 3, ALE, PSEN <sup>3</sup>	$I_{OH} = -60\mu\text{A}$ $I_{OH} = -30\mu\text{A}$ $I_{OH} = -10\mu\text{A}$	$V_{CC} - 1.5$ $V_{CC} - 0.7$ $V_{CC} - 0.3$			V V V
$V_{OH1}$	Output high voltage (port 0 in external bus mode), ALE <sup>10</sup> , PSEN <sup>3</sup>	$I_{OH} = -7.0\text{mA}$ $I_{OH} = -3.2\text{mA}$ $I_{OH} = -200\mu\text{A}$	$V_{CC} - 1.5$ $V_{CC} - 0.7$ $V_{CC} - 0.3$			V V V
$I_{IL}$	Logical 0 input current, ports 1, 2, 3 <sup>7</sup>	$V_{IN} = 0.45\text{V}$			-50	$\mu\text{A}$
$I_{TL}$	Logical 1-to-0 transition current, ports 1, 2, 3 <sup>7</sup>	See note 4			-650	$\mu\text{A}$
$I_{LI}$	Input leakage current, port 0	$0.45 V_{IN} < V_{CC} - 0.3$			$\pm 10$	$\mu\text{A}$
$I_{CC}$	Power supply current: <sup>7</sup> Active mode @ 16MHz <sup>5</sup> Idle mode @ 16MHz Power-down mode	See note 6		15 3 10	25 5 75	mA mA $\mu\text{A}$
$R_{RST}$	Internal reset pull-down resistor		50		225	k $\Omega$
$C_{IO}$	Pin capacitance <sup>11</sup> (except EA)				15	pF

## NOTES:

- Typical ratings are not guaranteed. The values listed are at room temperature, 5V.
- Capacitive loading on ports 0 and 2 may cause spurious noise to be superimposed on the  $V_{OL}$ s of ALE and ports 1 and 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading  $> 100\text{pF}$ ), the noise pulse on the ALE pin may exceed 0.8V. In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.  $I_{OL}$  can exceed these conditions provided that no single output sinks more than 5mA and no more than two outputs exceed the test conditions.
- Capacitive loading on ports 0 and 2 may cause the  $V_{OH}$  on ALE and PSEN to momentarily fall below the  $0.9V_{CC}$  specification when the address bits are stabilizing.
- Pins of ports 1, 2 and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when  $V_{IN}$  is approximately 2V.
- $I_{CCMAX}$  at other frequencies is given by: Active mode:  $I_{CCMAX} = 1.50 \times \text{FREQ} + 8$ ; Idle mode:  $I_{CCMAX} = 0.14 \times \text{FREQ} + 2.31$ , where FREQ is the external oscillator frequency in MHz.  $I_{CCMAX}$  is given in mA. See Figure 8.
- See Figures 9 through 12 for  $I_{CC}$  test conditions.
- These values apply only to  $T_{amb} = 0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$ . For  $T_{amb} = -40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ , see table on previous page.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:  
Maximum  $I_{OL}$  per port pin: 15mA (\*NOTE: This is 85°C specification.)  
Maximum  $I_{OL}$  per 8-bit port: 26mA  
Maximum total  $I_{OL}$  for all outputs: 71mA  
If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
- ALE is tested to  $V_{OH1}$ , except when ALE is off then  $V_{OH}$  is the voltage specification.
- Pin capacitance is less than 25pF. Pin capacitance of ceramic package is less than 15pF (except EA it is 25pF).

## CMOS single-chip 8-bit microcontroller

## 83C51FA/87C51FA

## AC ELECTRICAL CHARACTERISTICS

$T_{amb} = 0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$  or  $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$  1, 2, 3

SYMBOL	FIGURE	PARAMETER	16MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
$1/t_{CLCL}$	1	Oscillator frequency			3.5	16	MHz
					3.5	24	MHz
$t_{LHLL}$	1	ALE pulse width	85		$2t_{CLCL}-40$		ns
$t_{AVLL}$	1	Address valid to ALE low	22		$t_{CLCL}-40$		ns
$t_{LLAX}$	1	Address hold after ALE low	32		$t_{CLCL}-30$		ns
$t_{LLIV}$	1	ALE low to valid instruction in		150		$4t_{CLCL}-100$	ns
$t_{LLPL}$	1	ALE low to PSEN low	32		$t_{CLCL}-30$		ns
$t_{PLPH}$	1	PSEN pulse width	142		$3t_{CLCL}-45$		ns
$t_{PLIV}$	1	PSEN low to valid instruction in		82		$3t_{CLCL}-105$	ns
$t_{PXIX}$	1	Input instruction hold after PSEN	0		0		ns
$t_{PXIZ}$	1	Input instruction float after PSEN		37		$t_{CLCL}-25$	ns
$t_{AVIV}$	1	Address to valid instruction in		207		$5t_{CLCL}-105$	ns
$t_{PLAZ}$	1	PSEN low to address float		10		10	ns
<b>Data Memory</b>							
$t_{RLRH}$	2, 3	RD pulse width	275		$6t_{CLCL}-100$		ns
$t_{WLWH}$	2, 3	WR pulse width	275		$6t_{CLCL}-100$		ns
$t_{RLDV}$	2, 3	RD low to valid data in		147		$5t_{CLCL}-165$	ns
$t_{RHDZ}$	2, 3	Data hold after RD	0		0		ns
$t_{RHDZ}$	2, 3	Data float after RD		65		$2t_{CLCL}-60$	ns
$t_{LLDV}$	2, 3	ALE low to valid data in		350		$8t_{CLCL}-150$	ns
$t_{AVDV}$	2, 3	Address to valid data in		397		$9t_{CLCL}-165$	ns
$t_{LLWL}$	2, 3	ALE low to RD or WR low	137	237	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
$t_{AVWL}$	2, 3	Address valid to WR low or RD low	175		$4t_{CLCL}-75$		ns
$t_{QVWX}$	2, 3	Data valid to WR transition	42		$t_{CLCL}-20$		ns
$t_{WHQX}$	2, 3	Data hold after WR	42		$t_{CLCL}-20$		ns
$t_{QVWH}$	3	Data valid to WR high	287		$7t_{CLCL}-150$		ns
$t_{RLAZ}$	2, 3	RD low to address float		0		0	ns
$t_{WHLH}$	2, 3	RD or WR high to ALE high	40	87	$t_{CLCL}-20$	$t_{CLCL}+25$	ns
<b>External Clock</b>							
$t_{CHCX}$	5	High time	12		20		ns
$t_{CLCX}$	5	Low time	12		20		ns
$t_{CLCH}$	5	Rise time		20		20	ns
$t_{CHCL}$	5	Fall time		20		20	ns
<b>Shift Register</b>							
$t_{XLXL}$	4	Serial port clock cycle time	1		$12t_{CLCL}$		$\mu\text{s}$
$t_{QVXH}$	4	Output data setup to clock rising edge	492		$10t_{CLCL}-133$		ns
$t_{XHGX}$	4	Output data hold after clock rising edge	8		$2t_{CLCL}-117$		ns
$t_{XHDX}$	4	Input data hold after clock rising edge	0		0		ns
$t_{XHDV}$	4	Clock rising edge to input data valid	492		$10t_{CLCL}-133$		ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- Interfacing the 83C51FA to devices with float times up to 45ns is permitted. This limited bus contention will not cause damage to Port 0 drivers.

## 83C51FA/87C51FA

## EXPLANATION OF THE AC SYMBOLS

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

A – Address

C – Clock

D – Input data

H – Logic level high

- Instruction (program memory contents)

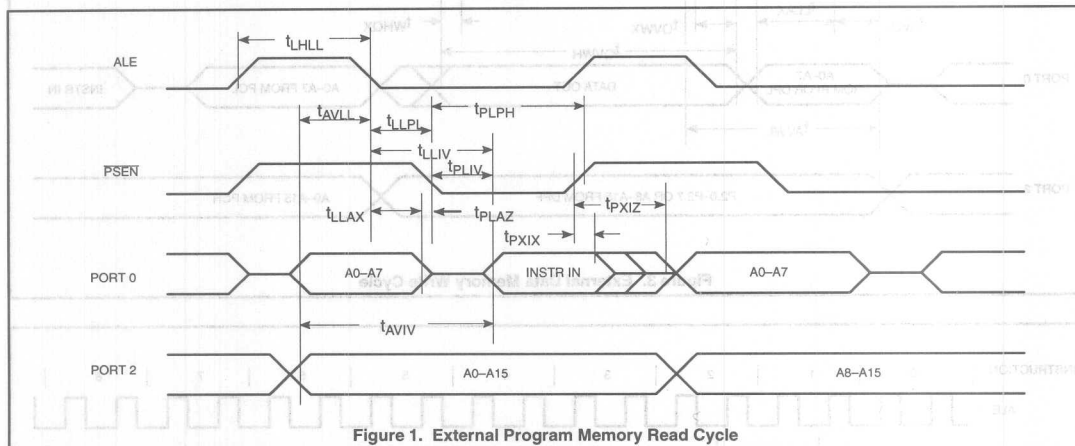
L – Logic level low, or ALE

P - PSEN

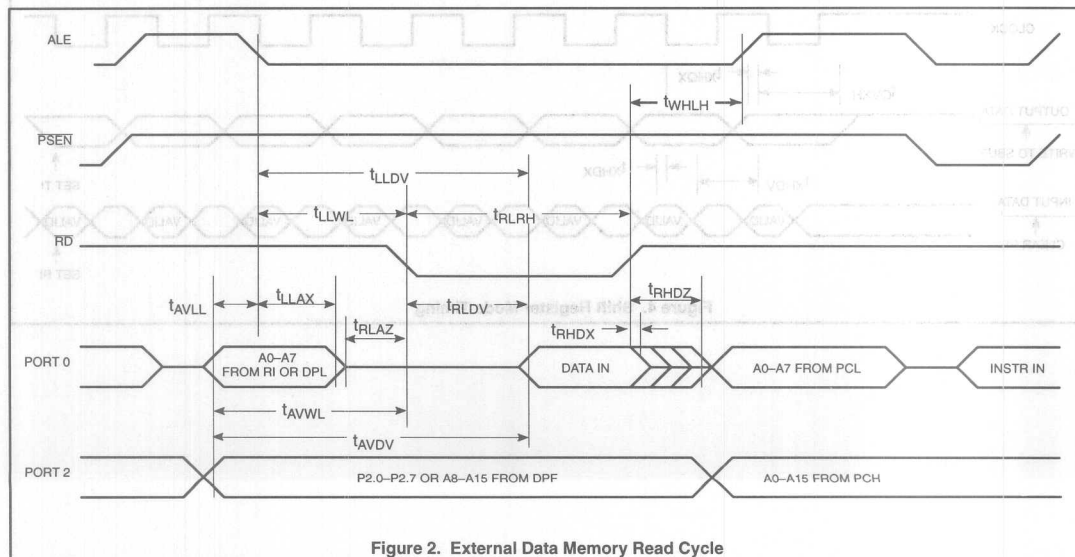
Z - Float

**Examples:**  $t_{AVLL}$  = Time for address valid to ALE low.

$t_{LLPL}$  = Time for ALE low to  $\overline{PSEN}$  low.



### Figure 1. External Program Memory Read Cycle



**Figure 2. External Data Memory Read Cycle**

## CMOS single-chip 8-bit microcontroller

83C51FA/87C51FA

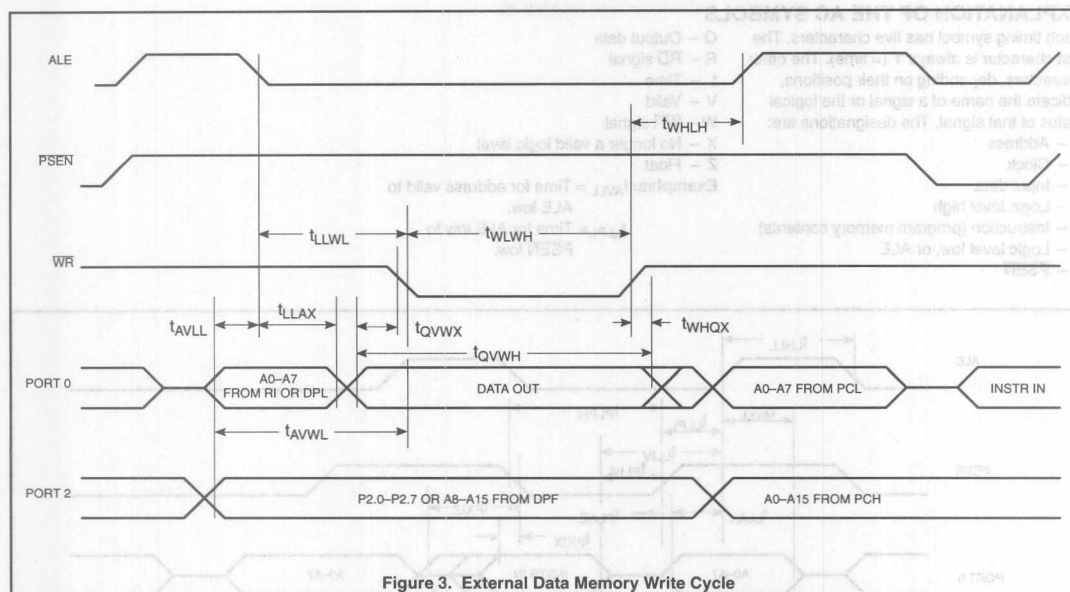


Figure 3. External Data Memory Write Cycle

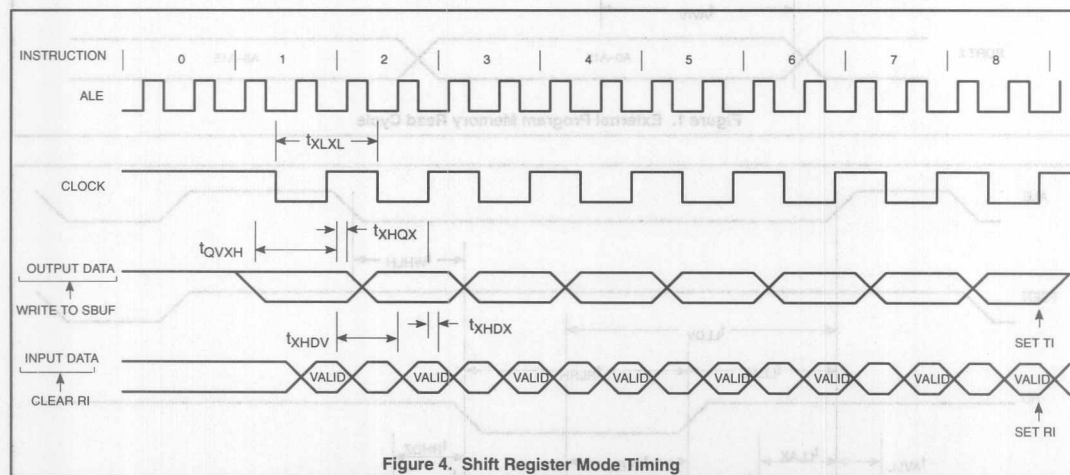
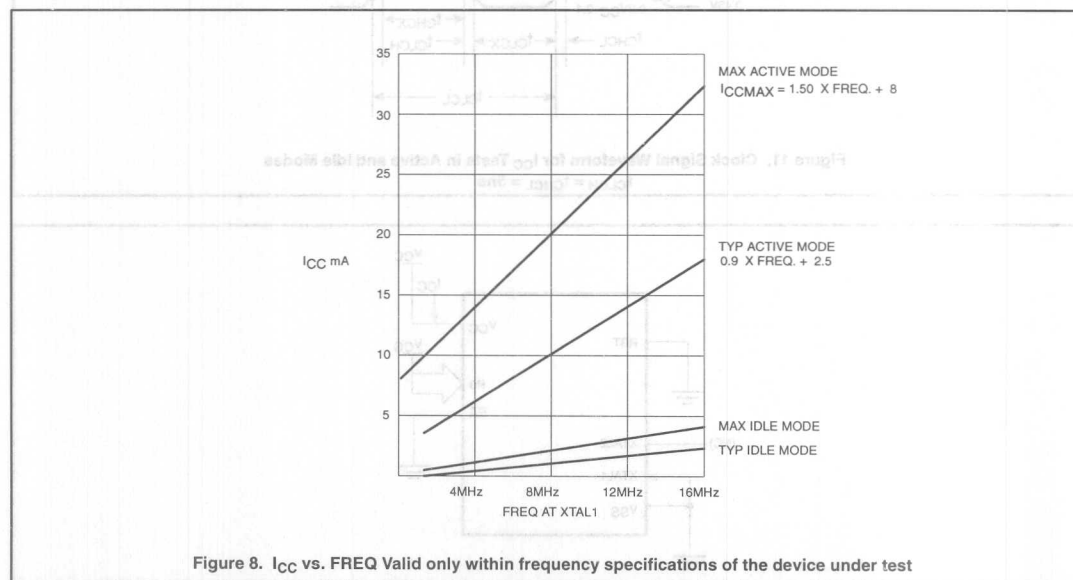
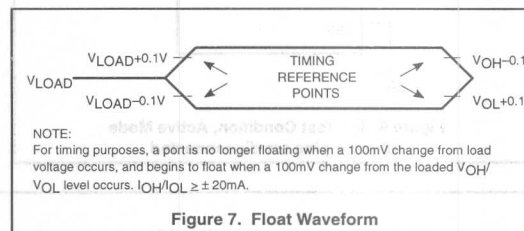
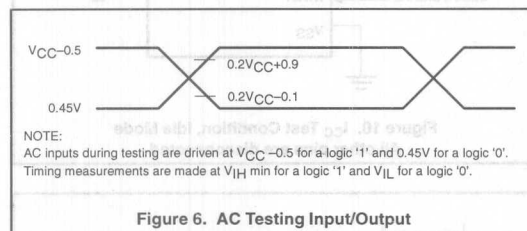
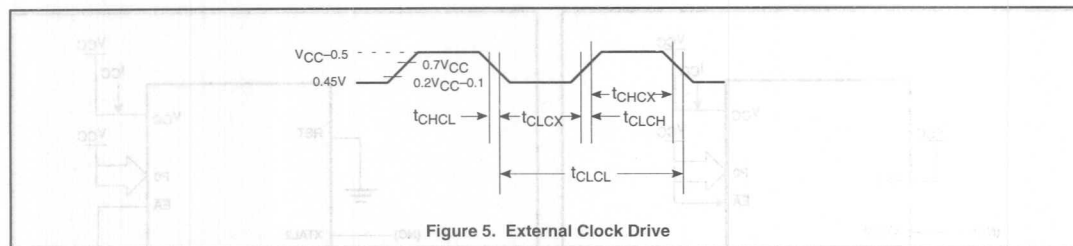


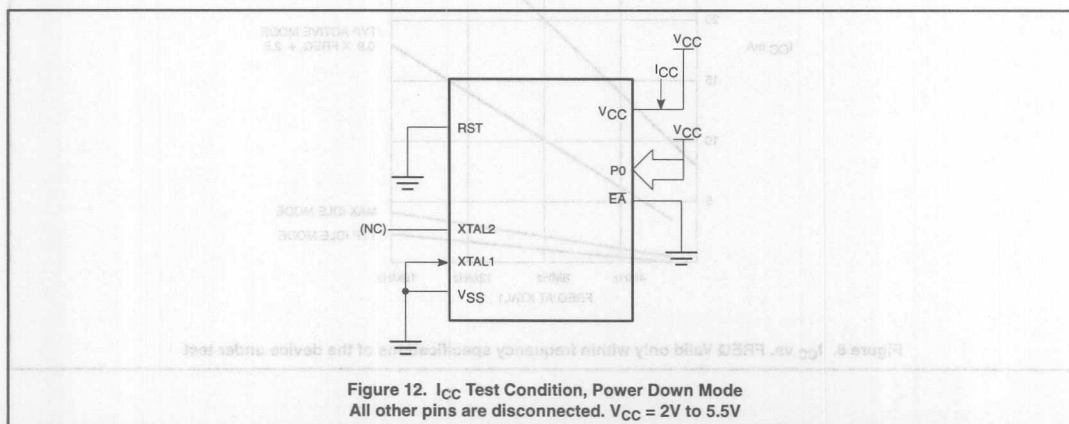
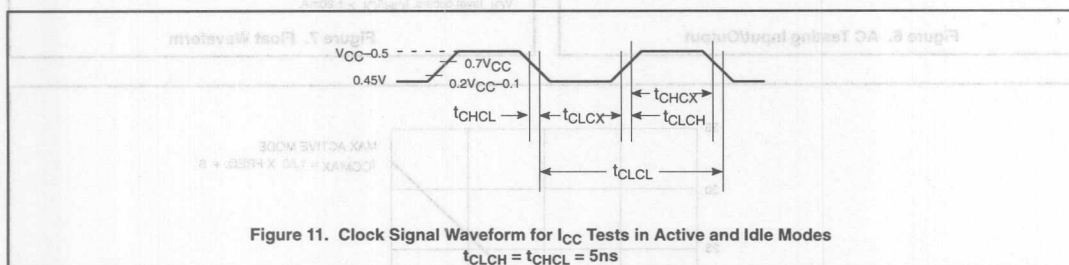
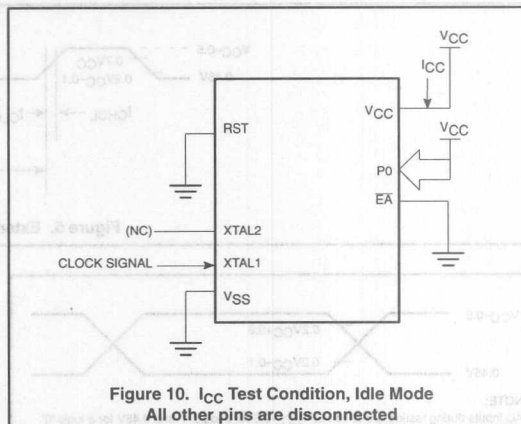
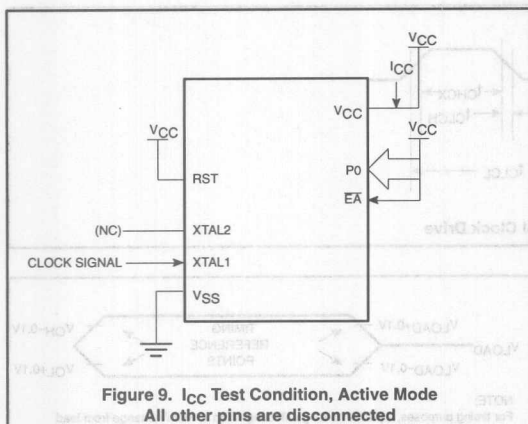
Figure 4. Shift Register Mode Timing

## CMOS single-chip 8-bit microcontroller 83C51FA/87C51FA





## CMOS single-chip 8-bit microcontroller 83C51FA/87C51FA



## CMOS single-chip 8-bit microcontroller 83C51FA/87C51FA

**EPROM CHARACTERISTICS**

The 87C51FA is programmed by using a modified Quick-Pulse Programming™ algorithm. It differs from older methods in the value used for  $V_{PP}$  (programming supply voltage) and in the width and number of the ALE/PROG pulses.

The 87C51FA contains two signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes identify the device as an 87C51FA manufactured by Philips.

Table 2 shows the logic levels for reading the signature byte, and for programming the program memory, the encryption table, and the security bits. The circuit configuration and waveforms for quick-pulse programming are shown in Figures 13 and 14. Figure 15 shows the circuit configuration for normal program memory verification.

**Quick-Pulse Programming**

The setup for microcontroller quick-pulse programming is shown in Figure 13. Note that the 87C51FA is running with a 3.5MHz to 12MHz oscillator. The reason the oscillator needs to be running is that the device is executing internal address and program data transfers.

The address of the EPROM location to be programmed is applied to ports 1 and 2, as shown in Figure 13. The code byte to be programmed into that location is applied to port 0. RST, PSEN and pins of ports 2 and 3 specified in Table 2 are held at the 'Program Code Data' levels indicated in Table 2. The ALE/PROG is pulsed low 15 to 25 times as shown in Figure 14.

To program the encryption table, repeat the 15 to 25 pulse programming sequence for addresses 0 through 1FH, using the 'Pgm Encryption Table' levels. Do not forget that after the encryption table is programmed, verification cycles will produce only encrypted data.

To program the security bits, repeat the 15 to 25 pulse programming sequence using the 'Pgm Security Bit' levels. After one security bit is programmed, further programming of the code memory and encryption table is disabled. However, the other security bit can still be programmed.

Note that the  $\overline{EA}/V_{PP}$  pin must not be allowed to go above the maximum specified  $V_{PP}$  level for any amount of time. Even a narrow glitch above that voltage can cause permanent damage to the device. The  $V_{PP}$  source should be well regulated and free of glitches and overshoot.

**Program Verification**

If security bit 2 has not been programmed, the on-chip program memory can be read out for program verification. The address of the program memory locations to be read is applied to ports 1 and 2 as shown in Figure 15. The other pins are held at the 'Verify Code Data' levels indicated in Table 2. The contents of the address location will be emitted on port 0. External pull-ups are required on port 0 for this operation.

If the encryption table has been programmed, the data presented at port 0 will be the exclusive NOR of the program byte with one of the encryption bytes. The user will have to know the encryption table contents in order to correctly decode the verification data. The encryption table itself cannot be read out.

**Reading the Signature Bytes**

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. The values are:

(030H) = 15H indicates manufactured by Philips

(031H) = B1H indicates 87C51FA

**Program/Verify Algorithms**

Any algorithm in agreement with the conditions listed in Table 2, and which satisfies the timing specifications, is suitable.

**Erase Characteristics**

Erase of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent erasure. **For this and secondary effects, it is recommended that an opaque label be placed over the window.** For elevated temperature or environments where solvents are being used, apply Kapton tape Fluorglas part number 2345-5, or equivalent.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 angstroms) to an integrated dose of at least 15W-s/cm<sup>2</sup>. Exposing the EPROM to an ultraviolet lamp of 12,000µW/cm<sup>2</sup> rating for 20 to 39 minutes, at a distance of about 1 inch, should be sufficient.

Erase leaves the array in an all 1s state.

**Table 2. EPROM Programming Modes**

MODE	RST	PSEN	ALE/PROG	$\overline{EA}/V_{PP}$	P2.7	P2.6	P3.7	P3.6
Read signature	1	0	1	1	0	0	0	0
Program code data	1	0	0*	$V_{PP}$	1	0	1	1
Verify code data	1	0	1	1	0	0	1	1
Pgm encryption table	1	0	0*	$V_{PP}$	1	0	1	0
Pgm security bit 1	1	0	0*	$V_{PP}$	1	1	1	1
Pgm security bit 2	1	0	0*	$V_{PP}$	1	1	0	0

**NOTES:**

1. '0' = Valid low for that pin, '1' = valid high for that pin.

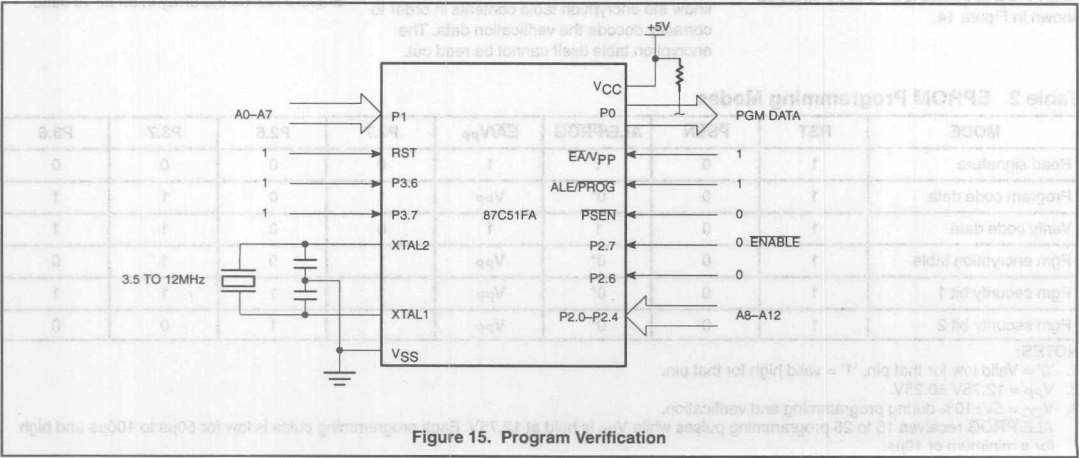
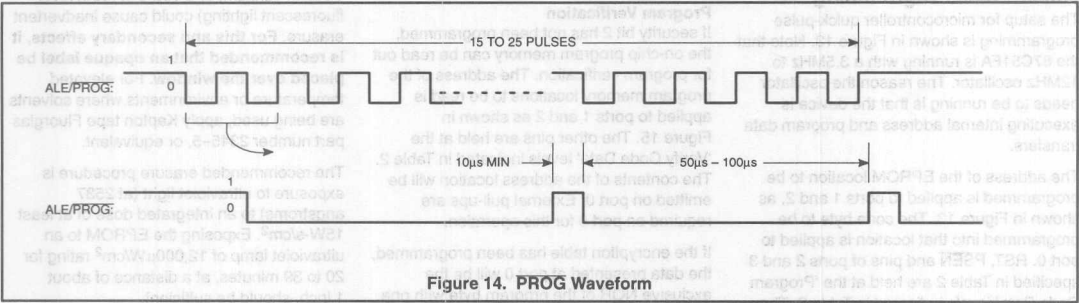
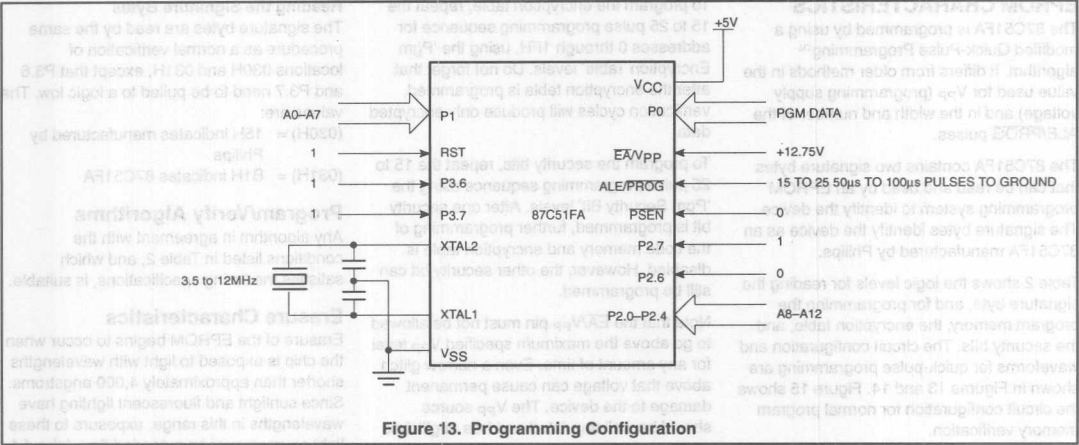
2.  $V_{PP} = 12.75V \pm 0.25V$ .

3.  $V_{CC} = 5V \pm 10\%$  during programming and verification.

\* ALE/PROG receives 15 to 25 programming pulses while  $V_{PP}$  is held at 12.75V. Each programming pulse is low for 50µs to 100µs and high for a minimum of 10µs.

™Trademark phrase of Intel Corporation.

CMOS single-chip 8-bit microcontroller 83C51FA/87C51FA

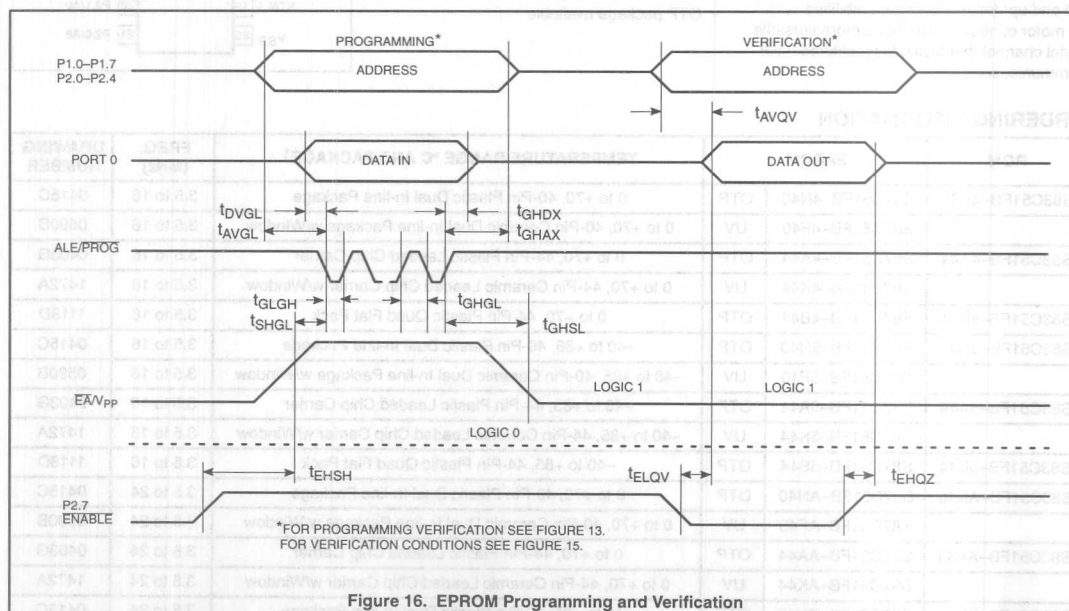


## CMOS single-chip 8-bit microcontroller 83C51FA/87C51FA

## EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

T<sub>amb</sub> = 21°C to +27°C, V<sub>CC</sub> = 5V±10%, V<sub>SS</sub> = 0V (See Figure 16)

SYMBOL	PARAMETER	MIN	MAX	UNIT
V <sub>PP</sub>	Programming supply voltage	12.5	13.0	V
I <sub>PP</sub>	Programming supply current		50	mA
1/t <sub>CLCL</sub>	Oscillator frequency	4	6	MHz
t <sub>AVGL</sub>	Address setup to PROG low	48t <sub>CLCL</sub>		
t <sub>GHAX</sub>	Address hold after PROG	48t <sub>CLCL</sub>		
t <sub>DVGL</sub>	Data setup to PROG low	48t <sub>CLCL</sub>		
t <sub>GHDX</sub>	Data hold after PROG	48t <sub>CLCL</sub>		
t <sub>EHSH</sub>	P2.7 (ENABLE) high to V <sub>PP</sub>	48t <sub>CLCL</sub>		
t <sub>SHGL</sub>	V <sub>PP</sub> setup to PROG low	10		µs
t <sub>GHSL</sub>	V <sub>PP</sub> hold after PROG	10		µs
t <sub>GLGH</sub>	PROG width	50	100	µs
t <sub>AVQV</sub>	Address to data valid		48t <sub>CLCL</sub>	
t <sub>ELQZ</sub>	ENABLE low to data valid		48t <sub>CLCL</sub>	
t <sub>EHQZ</sub>	Data float after ENABLE	0	48t <sub>CLCL</sub>	
t <sub>GHGL</sub>	PROG high to PROG low	10		µs



# CMOS single-chip 8-bit microcontrollers 83C51FB/87C51FB

## DESCRIPTION

The 83C51FB/87C51FB (hereafter collectively called 8XC51FB) Single-Chip 8-Bit Microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 8XC51FB has the same instruction set as the 80C51.

This device provides architectural enhancements that make it applicable in a variety of applications for general control systems. The 87C51FB contains 16k × 8 EPROM memory, the 83C51FB contains 16k × 8 ROM memory, a volatile 256 × 8 read/write data memory, four 8-bit I/O ports, three 16-bit timer/event counters, a Programmable Counter Array (PCA), a multi-source, two-priority-level, nested interrupt structure, an enhanced UART and on-chip oscillator and timing circuits. For systems that require extra capability, the 87C51FB can be expanded using standard TTL compatible memories and logic.

Its added features make it an even more powerful microcontroller for applications that require pulse width modulation, high-speed I/O and up/down counting capabilities such as motor control. It also has a more versatile serial channel that facilitates multiprocessor communications.

## FEATURES

- 80C51 central processing unit
- 16k × 8 EPROM expandable externally to 64k bytes (87C51FB)
  - Quick Pulse programming algorithm
  - Two level program security system
- 16k × 8 ROM (83C51FB)
- 256 × 8 RAM, expandable externally to 64k bytes
- Three 16-bit timer/counters
  - T2 is an up/down counter
- Programmable Counter Array (PCA)
  - High speed output
  - Capture/compare
  - Pulse Width Modulator
  - Watchdog Timer
- Four 8-bit I/O ports
- Full-duplex enhanced UART
  - Framing error detection
  - Automatic address recognition
- Power control modes
  - Idle mode
  - Power-down mode
- Once (On Circuit Emulation) Mode
- Five package styles
- OTP package available

## PIN CONFIGURATIONS

T2/P1.0	1	40	V <sub>CC</sub>
T2EX/P1.1	2	39	P0.0/AD0
ECU/P1.2	3	38	P0.1/AD1
CEX0/P1.3	4	37	P0.2/AD2
CEX1/P1.4	5	36	P0.3/AD3
CEX2/P1.5	6	35	P0.4/AD4
CEX3/P1.6	7	34	P0.5/AD5
CEX4/P1.7	8	33	P0.6/AD6
RST	9	32	P0.7/AD7
RxD/P3.0	10	31	E <sub>A</sub> /V <sub>PP</sub>
TxD/P3.1	11	30	ALE/PROG
INT0/P3.2	12	29	PSEN
INT1/P3.3	13	28	P2.7/A15
T0/P3.4	14	27	P2.6/A14
T1/P3.5	15	26	P2.5/A13
WR/P3.6	16	25	P2.4/A12
RD/P3.7	17	24	P2.3/A11
XTAL2	18	23	P2.2/A10
XTAL1	19	22	P2.1/A9
VSS	20	21	P2.0/A8

## ORDERING INFORMATION

ROM	EPROM		TEMPERATURE RANGE °C AND PACKAGE <sup>1</sup>	FREQ. (MHz)	DRAWING NUMBER
S83C51FB-4N40	S87C51FB-4N40	OTP	0 to +70, 40-Pin Plastic Dual In-line Package	3.5 to 16	0415C
	S87C51FB-4F40	UV	0 to +70, 40-Pin Ceramic Dual In-line Package w/Window	3.5 to 16	0590B
S83C51FB-4A44	S87C51FB-4A44	OTP	0 to +70, 44-Pin Plastic Leaded Chip Carrier	3.5 to 16	0403G
	S87C51FB-4K44	UV	0 to +70, 44-Pin Ceramic Leaded Chip Carrier w/Window	3.5 to 16	1472A
S83C51FB-4B44	S87C51FB-4B44	OTP	0 to +70, 44-Pin Plastic Quad Flat Pack	3.5 to 16	1118D
S83C51FB-5N40	S87C51FB-5N40	OTP	-40 to +85, 40-Pin Plastic Dual In-line Package	3.5 to 16	0415C
	S87C51FB-5F40	UV	-40 to +85, 40-Pin Ceramic Dual In-line Package w/Window	3.5 to 16	0590B
S83C51FB-5A44	S87C51FB-5A44	OTP	-40 to +85, 44-Pin Plastic Leaded Chip Carrier	3.5 to 16	0403G
	S87C51FB-5K44	UV	-40 to +85, 44-Pin Ceramic Leaded Chip Carrier w/Window	3.5 to 16	1472A
S83C51FB-5B44	S87C51FB-5B44	OTP	-40 to +85, 44-Pin Plastic Quad Flat Pack	3.5 to 16	1118D
S83C51FB-AN40	S87C51FB-AN40	OTP	0 to +70, 40-Pin Plastic Dual In-line Package	3.5 to 24	0415C
	S87C51FB-AF40	UV	0 to +70, 40-Pin Ceramic Dual In-line Package w/Window	3.5 to 24	0590B
S83C51FB-AA44	S87C51FB-AA44	OTP	0 to +70, 44-Pin Plastic Leaded Chip Carrier	3.5 to 24	0403G
	S87C51FB-AK44	UV	0 to +70, 44-Pin Ceramic Leaded Chip Carrier w/Window	3.5 to 24	1472A
S83C51FB-BN40	S87C51FB-BN40	OTP	-40 to +85, 40-Pin Plastic Dual In-line Package	3.5 to 24	0415C
	S87C51FB-BF40	UV	-40 to +85, 40-Pin Ceramic Dual In-line Package w/Window	3.5 to 24	0590B
S83C51FB-BA44	S87C51FB-BA44	OTP	-40 to +85, 44-Pin Plastic Leaded Chip Carrier	3.5 to 24	0403G
	S87C51FB-BK44	UV	-40 to +85, 44-Pin Ceramic Leaded Chip Carrier w/Window	3.5 to 24	1472A

### NOTE:

1. OTP = One Time Programmable EPROM. UV = Erasable EPROM.

## CMOS single-chip 8-bit microcontrollers 83C51FB/87C51FB



## CMOS single-chip 8-bit microcontrollers 83C51FB/87C51FB

Table 1. 8XC51FB Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB							LSB	
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
AUXR#	Auxiliary	8EH	—	—	—	—	—	—	—	AO	xxxxxx0B
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
CCAP0H#	Module 0 Capture High	FAH									xxxxxxxB
CCAP1H#	Module 1 Capture High	FBH									xxxxxxxB
CCAP2H#	Module 2 Capture High	FCH									xxxxxxxB
CCAP3H#	Module 3 Capture High	FDH									xxxxxxxB
CCAP4H#	Module 4 Capture High	FEH									xxxxxxxB
CCAP0L#	Module 0 Capture Low	EAH									xxxxxxxB
CCAP1L#	Module 1 Capture Low	EBH									xxxxxxxB
CCAP2L#	Module 2 Capture Low	ECH									xxxxxxxB
CCAP3L#	Module 3 Capture Low	EDH									xxxxxxxB
CCAP4L#	Module 4 Capture Low	EEH									xxxxxxxB
CCAPM0#	Module 0 Mode	DAH	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM1#	Module 1 Mode	DBH	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM2#	Module 2 Mode	DCH	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM3#	Module 3 Mode	DDH	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM4#	Module 4 Mode	DEH	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCON*#	PCA Counter Control	D8H	DF	DE	DD	DC	DB	DA	D9	D8	00x00000B
CH#	PCA Counter High	F9H	CF	CR	—	CCF4	CCF3	CCF2	CCF1	CCF0	00H
CL#	PCA Counter Low	E9H									00H
CMOD#	PCA Counter Mode	D9H	CIDL	WDTE	—	—	—	CPS1	CPS0	ECF	00xxx000B
DPTR:	Data Pointer (2 bytes)										
DPH	Data Pointer High	83H									00H
DPL	Data Pointer Low	82H									00H
IE*	Interrupt Enable	A8H	AF	AE	AD	AC	AB	AA	A9	A8	00H
			EA	EC	ET2	ES	ET1	EX1	ET0	EX0	
			BF	BE	BD	BC	BB	BA	B9	B8	
IP*	Interrupt Priority	B8H	—	PPC	PT2	PS	PT1	PX1	PT0	PX0	x0000000B
			87	86	85	84	83	82	81	80	
P0*	Port 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
			97	96	95	94	93	92	91	90	
P1*	Port 1	90H	CEX4	CEX3	CEX2	CEX1	CEX0	EX1	T2EX	T2	FFH
			A7	A6	A5	A4	A3	A2	A1	A0	
P2*	Port 2	A0H	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8	FFH
			B7	B6	B5	B4	B3	B2	B1	B0	
P3*	Port 3	B0H	RD	WR	T1	T0	INT1	INT0	TxD	RxD	FFH
PCON	Power Control	87H	SMOD1	SMOD0	—	POF <sup>1</sup>	GF1	GF0	PD	IDL	00xxxx00B

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.

## CMOS single-chip 8-bit microcontrollers 83C51FB/87C51FB

Table 1. 83C51FB Special Function Registers (Continued)

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB				LSB				
PSW*	Program Status Word	D0H	D7	D6	D5	D4	D3	D2	D1	D0	00H
	RACAP2H#	CBH	CY	AC	F0	RS1	RS0	OV	—	P	00H
	RACAP2L#	CAH									00H
SADDR#	Slave Address	A9H									00H
SADEN#	Slave Address Mask	B9H									00H
SBUF	Serial Data Buffer	99H									xxxxxxxxB
SCON*	Serial Control	98H	9F	9E	9D	9C	9B	9A	99	98	00H
SP	Stack Pointer	81H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	07H
TCON*	Timer Control	88H	8F	8E	8D	8C	8B	8A	89	88	00H
			TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
			CF	CE	CD	CC	CB	CA	C9	C8	00H
T2CON*	Timer 2 Control	C8H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	00H
T2MOD#	Timer 2 Mode Control	C9H	—	—	—	—	—	—	—	DCEN	xxxxxxx0B
TH0	Timer High 0	8CH									00H
TH1	Timer High 1	8DH									00H
TH2#	Timer High 2	CDH									00H
TL0	Timer Low 0	8AH									00H
TL1	Timer Low 1	8BH									00H
TL2#	Timer Low 2	CCH									00H
TMOD	Timer Mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H
			C7	C6	C5	C4	C3	C2	C1	C0	00H

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.

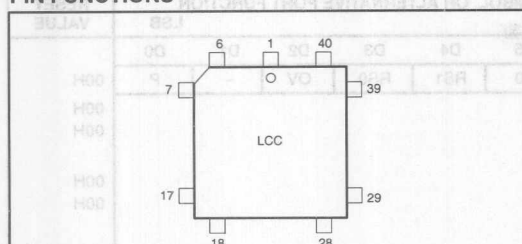
1. Reset value depends on reset source.

NAME AND FUNCTION	TYPE	OFF	LOC	DIR	INITIAL
Ground: 0V reference.		10	20	1	00H
Power Supply: This is the power supply voltage for normal, idle, and power-down operation.		30	30	1	00H
Port 0: Port 0 is an open-drain, bidirectional I/O port. Port 0 pins have to be written to them first and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s. Port 0 also outputs the code bytes during program verification and receives code bytes during EPROM programming. External pull-ups are required during program verification.	I/O	37-30	37-30	1	00H
Port 1: Port 1 is an 8-bit bidirectional I/O port with internal pull-up, except P1.6 and P1.7, which are open drain. Port 1 pins that have to be written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current (maximum of the internal pull-up, 100 $\mu$ A) to the external circuitry. Port 1 also receives the low-order address bytes during program memory verification. Alternate functions include: T2 (P1.0): Timer/Counter 2 external count input (pin 10) T2EX (P1.1): Timer/Counter 2 external capture/division input EOT (P1.2): External clock input to the PCA CEX0 (P1.3): Capture/Compare External I/O for PCA module 0 CEX1 (P1.4): Capture/Compare External I/O for PCA module 1 CEX2 (P1.5): Capture/Compare External I/O for PCA module 2 CEX3 (P1.6): Capture/Compare External I/O for PCA module 3 CEX4 (P1.7): Capture/Compare External I/O for PCA module 4	I/O	40-44	40-44	1	00H
		45	45	1	00H
		46	46	1	00H
		47	47	1	00H
		48	48	1	00H
		49	49	1	00H
		50	50	1	00H
		51	51	1	00H
		52	52	1	00H
		53	53	1	00H
		54	54	1	00H
		55	55	1	00H
		56	56	1	00H
		57	57	1	00H
		58	58	1	00H
		59	59	1	00H
		60	60	1	00H
		61	61	1	00H
		62	62	1	00H
		63	63	1	00H
		64	64	1	00H
		65	65	1	00H
		66	66	1	00H
		67	67	1	00H
		68	68	1	00H
		69	69	1	00H
		70	70	1	00H
		71	71	1	00H
		72	72	1	00H
		73	73	1	00H
		74	74	1	00H
		75	75	1	00H
		76	76	1	00H
		77	77	1	00H
		78	78	1	00H
		79	79	1	00H
		80	80	1	00H
		81	81	1	00H
		82	82	1	00H
		83	83	1	00H
		84	84	1	00H
		85	85	1	00H
		86	86	1	00H
		87	87	1	00H
		88	88	1	00H
		89	89	1	00H
		90	90	1	00H
		91	91	1	00H
		92	92	1	00H
		93	93	1	00H
		94	94	1	00H
		95	95	1	00H
		96	96	1	00H
		97	97	1	00H
		98	98	1	00H
		99	99	1	00H
		100	100	1	00H
		101	101	1	00H
		102	102	1	00H
		103	103	1	00H
		104	104	1	00H
		105	105	1	00H
		106	106	1	00H
		107	107	1	00H
		108	108	1	00H
		109	109	1	00H
		110	110	1	00H
		111	111	1	00H
		112	112	1	00H
		113	113	1	00H
		114	114	1	00H
		115	115	1	00H
		116	116	1	00H
		117	117	1	00H
		118	118	1	00H
		119	119	1	00H
		120	120	1	00H
		121	121	1	00H
		122	122	1	00H
		123	123	1	00H
		124	124	1	00H
		125	125	1	00H
		126	126	1	00H
		127	127	1	00H

## CMOS single-chip 8-bit microcontrollers 83C51FB/87C51FB

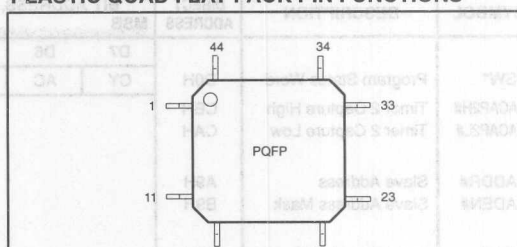
## CERAMIC AND PLASTIC LEADED CHIP CARRIER

## PIN FUNCTIONS



Pin	Function	Pin	Function
1	NC	23	NC
2	P1.0/T2	24	P2.0/A8
3	P1.1/T2EX	25	P2.1/A9
4	P1.2/ECI	26	P2.2/A10
5	P1.3/CEX0	27	P2.3/A11
6	P1.4/CEX1	28	P2.4/A12
7	P1.5/CEX2	29	P2.5/A13
8	P1.6/CEX3	30	P2.6/A14
9	P1.7/CEX4	31	P2.7/A15
10	RST	32	PSEN
11	P3.0/RxD	33	ALE/PROG
12	NC	34	NC
13	P3.1/TxD	35	EA/Vpp
14	P3.2/INT0	36	P0.7/AD7
15	P3.3/INT1	37	P0.6/AD6
16	P3.4/T0	38	P0.5/AD5
17	P3.5/T1	39	P0.4/AD4
18	P3.6/WR	40	P0.3/AD3
19	P3.7/RD	41	P0.2/AD2
20	XTAL2	42	P0.1/AD1
21	XTAL1	43	P0.0/AD0
22	VSS	44	VCC

## PLASTIC QUAD FLAT PACK PIN FUNCTIONS



Pin	Function	Pin	Function
1	P1.5/CEX2	23	P2.5/A13
2	P1.6/CEX3	24	P2.6/A14
3	P1.7/CEX4	25	P2.7/A15
4	RST	26	PSEN
5	P3.0/RxD	27	ALE/PROG
6	NC	28	NC
7	P3.1/TxD	29	EA/Vpp
8	P3.2/INT0	30	P0.7/AD7
9	P3.3/INT1	31	P0.6/AD6
10	P3.4/T0	32	P0.5/AD5
11	P3.5/T1	33	P0.4/AD4
12	P3.6/WR	34	P0.3/AD3
13	P3.7/RD	35	P0.2/AD2
14	XTAL2	36	P0.1/AD1
15	XTAL1	37	P0.0/AD0
16	VSS	38	VCC
17	NC	39	NC
18	P2.0/A8	40	P1.0/T2
19	P2.1/A9	41	P1.1/T2EX
20	P2.2/A10	42	P1.2/ECI
21	P2.3/A11	43	P1.3/CEX0
22	P2.4/A12	44	P1.4/CEX1

## PIN DESCRIPTIONS

MNEMONIC	PIN NUMBER			TYPE	NAME AND FUNCTION
	DIP	LCC	QFP		
VSS	20	22	16	I	Ground: 0V reference.
VCC	40	44	38	I	<b>Power Supply:</b> This is the power supply voltage for normal, idle, and power-down operation.
P0.0–0.7	39–32	43–36	37–30	I/O	<b>Port 0:</b> Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s. Port 0 also outputs the code bytes during program verification and receives code bytes during EPROM programming. External pull-ups are required during program verification.
P1.0–P1.7	1–8	2–9	40–44, 1–3	I/O	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pull-ups, except P1.6 and P1.7 which are open drain. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 1 also receives the low-order address byte during program memory verification. Alternate functions include: <ul style="list-style-type: none"> <li><b>T2 (P1.0):</b> Timer/Counter 2 external count input/Clockout</li> <li><b>T2EX (P1.1):</b> Timer/Counter 2 Reload/Capture/Direction Control</li> <li><b>ECI (P1.2):</b> External Clock Input to the PCA</li> <li><b>CEX0 (P1.3):</b> Capture/Compare External I/O for PCA module 0</li> <li><b>CEX1 (P1.4):</b> Capture/Compare External I/O for PCA module 1</li> <li><b>CEX2 (P1.5):</b> Capture/Compare External I/O for PCA module 2</li> <li><b>CEX3 (P1.6):</b> Capture/Compare External I/O for PCA module 3</li> <li><b>CEX4 (P1.7):</b> Capture/Compare External I/O for PCA module 4</li> </ul>
	1	2	40	I	
	2	3	41	I	
	3	4	42	I	
	4	5	43	I/O	
	5	6	44	I/O	
	6	7	1	I/O	
	7	8	2	I/O	
	8	9	3	I/O	

## CMOS single-chip 8-bit microcontrollers 83C51FB/87C51FB

## PIN DESCRIPTIONS (Continued)

MNEMONIC	PIN NUMBER			TYPE	NAME AND FUNCTION
	DIP	LCC	QFP		
P2.0–P2.7	21–28	24–31	18–25	I/O	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: $I_{IL}$ ). Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register. Some Port 2 pins receive the high order address bits during EPROM programming and verification.
P3.0–P3.7	10–17	11, 13–19	5, 7–13	I/O	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: $I_{IL}$ ). Port 3 also serves the special features of the 80C51 family, as listed below: <b>RxD (P3.0):</b> Serial input port <b>TxD (P3.1):</b> Serial output port <b>INT0 (P3.2):</b> External interrupt <b>INT1 (P3.3):</b> External interrupt <b>T0 (P3.4):</b> Timer 0 external input <b>T1 (P3.5):</b> Timer 1 external input <b>WR (P3.6):</b> External data memory write strobe <b>RD (P3.7):</b> External data memory read strobe
RST	9	10	4	I	<b>Reset:</b> A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to $V_{SS}$ permits a power-on reset using only an external capacitor to $V_{CC}$ .
ALE/PROG	30	33	27	I/O	<b>Address Latch Enable/Program Pulse:</b> Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory. This pin is also the program pulse input (PROG) during EPROM programming.
PSEN	29	32	26	O	<b>Program Store Enable:</b> The read strobe to external program memory. When the 87C51FB is executing code from the external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory. PSEN is not activated during fetches from internal program memory.
EA/V <sub>PP</sub>	31	35	29	I	<b>External Access Enable/Programming Supply Voltage:</b> EA must be externally held low to enable the device to fetch code from external program memory locations 0000H and 3FFFH. If EA is held high, the device executes from internal program memory unless the program counter contains an address greater than 3FFFH. This pin also receives the 12.75V programming supply voltage ( $V_{PP}$ ) during EPROM programming. If security bit 1 is programmed, EA will be internally latched on Reset.
XTAL1	19	21	15	I	<b>Crystal 1:</b> Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	20	14	O	<b>Crystal 2:</b> Output from the inverting oscillator amplifier.

**NOTE:**

To avoid "latch-up" effect at power-on, the voltage on any pin at any time must not be higher than  $V_{CC} + 0.5V$  or  $V_{SS} - 0.5V$ , respectively.

**TIMER 2**

This is a 16-bit up or down counter, which can be operated as either a timer or event counter. It can be operated in one of three different modes (autoreload, capture or as the baud rate generator for the UART).

In the autoreload mode the Timer can be set to count up or down by setting or clearing the bit DCEN in the T2CON Special Function Register. The SFR's RCAP2H and RCAP2L are used to reload the Timer upon overflow or a 1-to-0 transition on the T2EX input (P1.1).

In the Capture mode Timer 2 can either set TF2 and generate an interrupt or capture its value. To capture Timer 2 in response to a 1-to-0 transition on the T2EX input, the EXEN2 bit in the T2CON must be set. Timer 2 is then captured in SFR's RCAP2H and RCAP2L.

As the baud rate generator, Timer 2 is selected by setting TCLK and/or RCLK in T2CON. As the baud rate generator Timer 2 is incremented at  $1/2$  the oscillator frequency.

**POWER OFF FLAG**

The Power Off Flag (POF) is set by on-chip circuitry when the  $V_{CC}$  level on the 87C51FB rises from 0 to 5V. The POF bit can be set or cleared by software allowing a user to determine if the reset is the result of a power-on or a warm start after powerdown. The  $V_{CC}$  level must remain above 3V for the POF to remain unaffected by the  $V_{CC}$  level.

CMOS single-chip 8-bit microcontrollers 83C51FB/87C51FB

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

Reset

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. To insure a good power-on reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few milliseconds) plus two machine cycles. At power-on, the voltage on VCC and RST must come up at the same time for a proper start-up.

Idle Mode

In the idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle

mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

Power-Down Mode

To save even more power, a Power Down mode can be invoked by software. In this mode, the oscillator is stopped and the instruction that invoked Power Down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power Down mode is terminated.

On the 87C51FB either a hardware reset or external interrupt can use an exit from Power Down. Reset redefines all the SFRs but does not change the on-chip RAM. An external interrupt allows both the SFRs and the on-chip RAM to retain their values.

To properly terminate Power Down the reset or external interrupt should not be executed before VCC is restored to its normal operating level and must be held active long enough for the oscillator to restart and stabilize (normally less than 10ms).

With an external interrupt, INT0 and INT1 must be enabled and configured as level-sensitive. Holding the pin low restarts the oscillator but bringing the pin back high completes the exit. Once the interrupt is serviced, the next instruction to be executed after RETI will be the one following the

instruction that put the device into Power Down.

Design Consideration

- When the idle mode is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

ONCE™ Mode

The ONCE ("On-Circuit Emulation") Mode facilitates testing and debugging of systems using the 87C51FB without the 87C51FB having to be removed from the circuit. The ONCE Mode is invoked by:

- Pull ALE low while the device is in reset and PSEN is high;
- Hold ALE low as RST is deactivated.

While the device is in ONCE Mode, the Port 0 pins go into a float state, and the other port pins and ALE and PSEN are weakly pulled high. The oscillator circuit remains active. While the 87C51FB is in this mode, an emulator or test CPU can be used to drive the circuit. Normal operation is restored when a normal reset is applied.

Table 2. External Pin Status During Idle and Power-Down Mode

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data



## CMOS single-chip 8-bit microcontrollers 83C51FB/87C51FB

**Programmable Counter Array (PCA)**

The Programmable Counter Array is a special Timer that has five 16-bit capture/compare modules associated with it. Each of the modules can be programmed to operate in one of four modes: rising and/or falling edge capture, software timer, high-speed output, or pulse width modulator. Each module has a pin associated with it in port 1. Module 0 is connected to P1.3(CEX0), module 1 to P1.4(CEX1), etc. The basic PCA configuration is shown in Figure 1.

The PCA timer is a common time base for all five modules and can be programmed to run at: 1/12 the oscillator frequency, 1/4 the oscillator frequency, the Timer 0 overflow, or the input on the ECI pin (P1.2). The timer count source is determined from the CPS1 and CPS0 bits in the CMOD SFR as follows (see Figure 4):

CPS1	CPS0	PCA Timer Count Source
0	0	1/12 oscillator frequency
0	1	1/4 oscillator frequency
1	0	Timer 0 overflow
1	1	External Input at ECI pin

In the CMOD SFR are three additional bits associated with the PCA. They are CIDL which allows the PCA to stop during idle mode, WDTE which enables or disables the watchdog function on module 4, and ECF which when set causes an interrupt and the PCA overflow flag CF (in the CCON SFR) to

be set when the PCA timer overflows. These functions are shown in Figure 2.

The watchdog timer function is implemented in module 4 as implemented in other parts that have a PCA that are available on the market. However, if a watchdog timer is required in the target application, it is recommended to use the hardware watchdog timer that is implemented on the 8XC51FB separately from the PCA (see Figure 12).

The CCON SFR contains the run control bit for the PCA and the flags for the PCA timer (CF) and each module (refer to Figure 5). To run the PCA the CR bit (CCON.6) must be set by software. The PCA is shut off by clearing this bit. The CF bit (CCON.7) is set when the PCA counter overflows and an interrupt will be generated if the ECF bit in the CMOD register is set. The CF bit can only be cleared by software. Bits 0 through 4 of the CCON register are the flags for the modules (bit 0 for module 0, bit 1 for module 1, etc.) and are set by hardware when either a match or a capture occurs. These flags also can only be cleared by software. The PCA interrupt system shown in Figure 3.

Each module in the PCA has a special function register associated with it. These registers are: CCAPM0 for module 0, CCAPM1 for module 1, etc. (see Figure 6). The registers contain the bits that control the mode that each module will operate in. The ECCF bit (CCAPMn.0 where n=0, 1, 2, 3, or

4 depending on the module) enables the CCF flag in the CCON SFR to generate an interrupt when a match or compare occurs in the associated module. PWM (CCAPMn.1) enables the pulse width modulation mode. The TOG bit (CCAPMn.2) when set causes the CEX output associated with the module to toggle when there is a match between the PCA counter and the module's capture/compare register. The match bit MAT (CCAPMn.3) when set will cause the CCFn bit in the CCON register to be set when there is a match between the PCA counter and the module's capture/compare register.

The next two bits CAPN (CCAPMn.4) and CAPP (CCAPMn.5) determine the edge that a capture input will be active on. The CAPN bit enables the negative edge, and the CAPP bit enables the positive edge. If both bits are set both edges will be enabled and a capture will occur for either transition. The last bit in the register ECOM (CCAPMn.6) when set enables the comparator function. Figure 7 shows the CCAPMn settings for the various PCA functions.

There are two additional registers associated with each of the PCA modules. They are CCAPnH and CCAPnL and these are the registers that store the 16-bit count when a capture occurs or a compare should occur. When a module is used in the PWM mode these registers are used to control the duty cycle of the output.

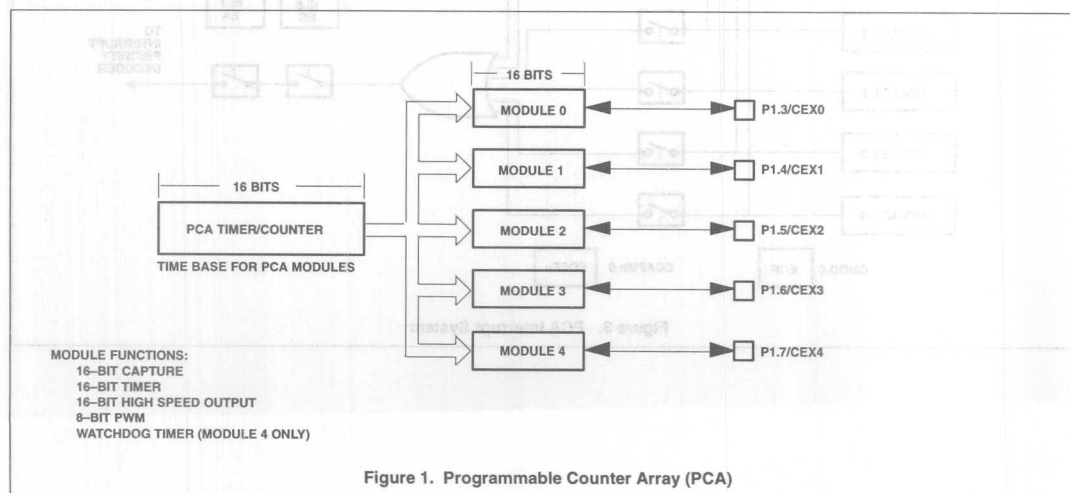
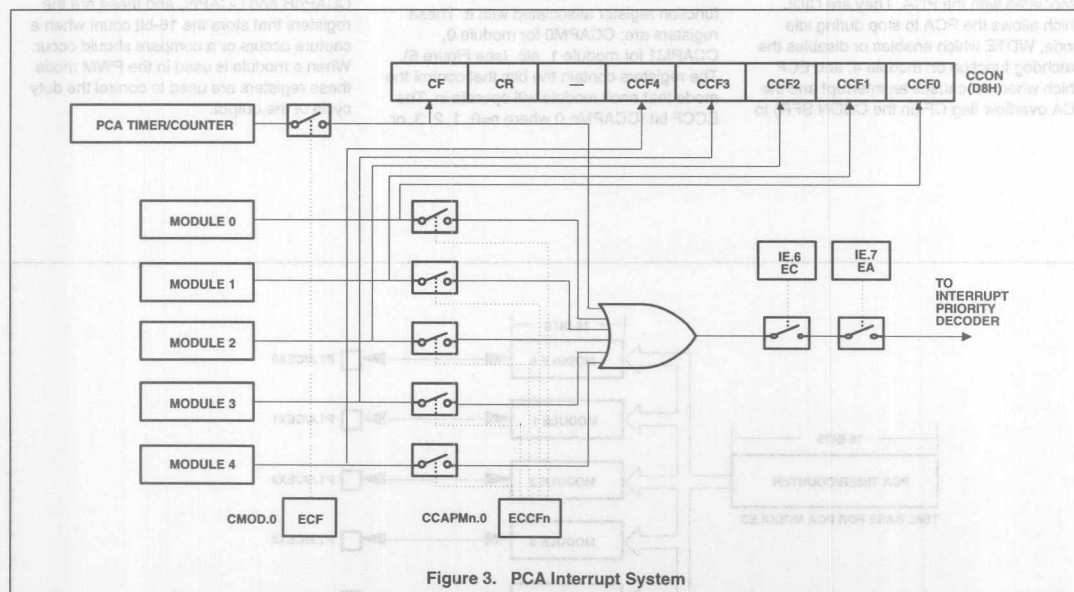
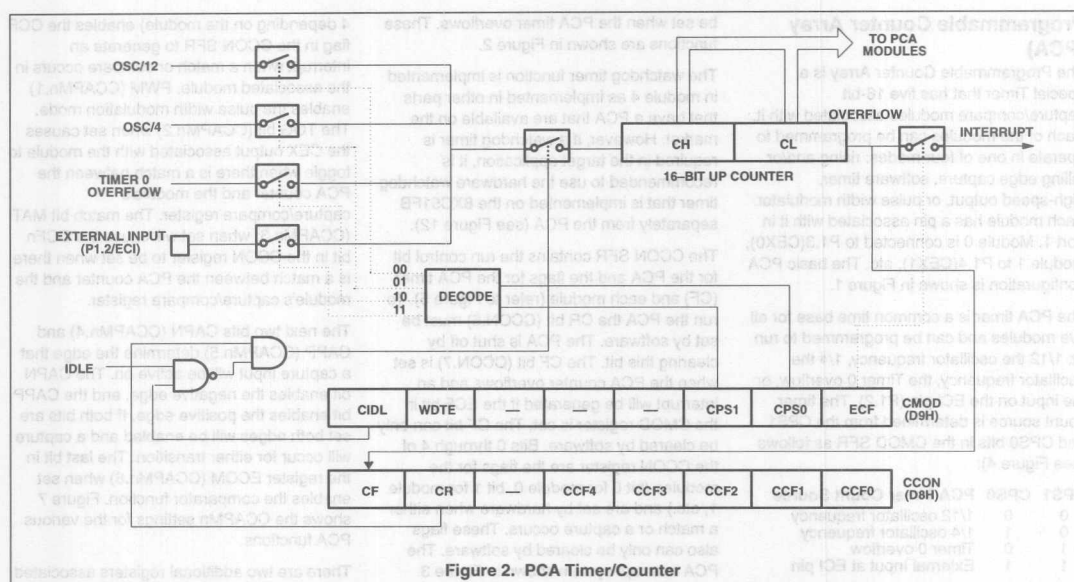


Figure 1. Programmable Counter Array (PCA)



## CMOS single-chip 8-bit microcontrollers 83C51FB/87C51FB



## CMOS single-chip 8-bit microcontrollers

## 83C51FB/87C51FB

CMOD Address = 0D9H		Reset Value = 00XX X000B							
Bit Addressable									
		CIDL	WDTE	-	-	-	CPS1	CPS0	ECF
Bit:	7	6	5	4	3	2	1	0	
Symbol	Function								
CIDL	Counter Idle control: CIDL = 0 programs the PCA Counter to continue functioning during idle Mode. CIDL = 1 programs it to be gated off during idle.								
WDTE	Watchdog Timer Enable: WDTE = 0 disables Watchdog Timer function on PCA Module 4. WDTE = 1 enables it.								
-	Not implemented, reserved for future use.*								
CPS1	PCA Count Pulse Select bit 1.								
CPS0	PCA Count Pulse Select bit 0.								
	CPS1	CPS0	Selected PCA Input**						
	0	0	0	Internal clock, Fosc ÷ 12					
	0	1	1	Internal clock, Fosc ÷ 4					
	1	0	2	Timer 0 overflow					
	1	1	3	External clock at ECI/P1.2 pin (max. rate = Fosc ÷ 8)					
ECF	PCA Enable Counter Overflow interrupt: ECF = 1 enables CF bit in CCON to generate an interrupt. ECF = 0 disables that function of CF.								
NOTE:									
*User software should not write 1s to reserved bits. These bits may be used in future 8051 family products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1. The value read from a reserved bit is indeterminate.									
**Fosc = oscillator frequency									

Figure 4. CMOD: PCA Counter Mode Register

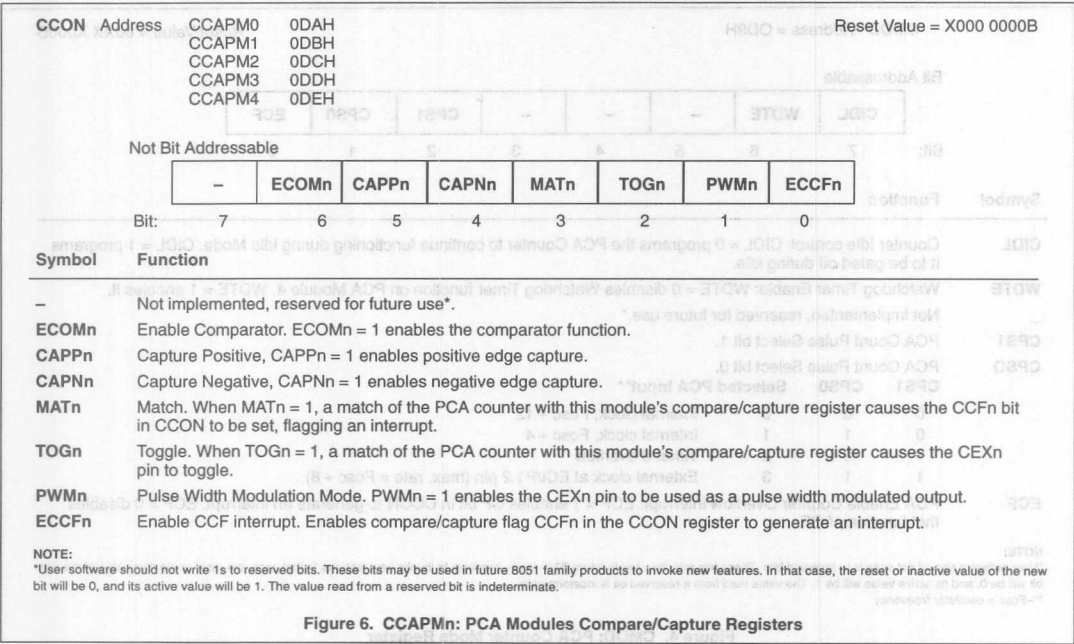
Figure 4. CMOD: PCA Counter Mode Register

CCON Address = 0D8H		Reset Value = 00X0 0000B						
Bit Addressable								
	CF	CR	—	CCF4	CCF3	CCF2	CCF1	CCF0
Bit:	7	6	5	4	3	2	1	0
Symbol	Function							
CF	PCA Counter Overflow flag. Set by hardware when the counter rolls over. CF flags an interrupt. If bit ECF in CMOD is set, CF may be set by either hardware or software but can only be cleared by software.							
CR	PCA Counter Run control bit. Set by software to turn the PCA counter on. Must be cleared by software to turn the PCA counter off.							
—	Not implemented, reserved for future use*.							
CCF4	PCA Module 4 interrupt flag. Set by hardware when a match or capture occurs. Must be cleared by software.							
CCF3	PCA Module 3 interrupt flag. Set by hardware when a match or capture occurs. Must be cleared by software.							
CCF2	PCA Module 2 interrupt flag. Set by hardware when a match or capture occurs. Must be cleared by software.							
CCF1	PCA Module 1 interrupt flag. Set by hardware when a match or capture occurs. Must be cleared by software.							
CCF0	PCA Module 0 interrupt flag. Set by hardware when a match or capture occurs. Must be cleared by software.							
NOTE:								
*User software should not write 1s to reserved bits. These bits may be used in future 8051 family products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1. The value read from a reserved bit is indeterminate.								

Figure 5. CCON: PCA Counter Control Register

Figure 5. CCON: PCA Counter Control Register

CMOS single-chip 8-bit microcontrollers 83C51FB/87C51FB



PCA Capture Mode

To use one of the PCA modules in the capture mode either one or both of the CCAPM bits CAPN and CAPP for that module must be set. The external CEX input for the module (on port 1) is sampled for a transition. When a valid transition occurs the PCA hardware loads the value of the PCA counter registers (CH and CL) into the module's capture registers (CCAPnL and CCAPnH). If the CCFn bit for the module in the CCON SFR and the ECCFn bit in the CCAPMn SFR are set then an interrupt will be generated. Refer to Figure 8.

16-bit Software Timer Mode

The PCA modules can be used as software timers by setting both the ECOM and MAT bits in the modules CCAPMn register. The PCA timer will be compared to the module's capture registers and when a match occurs an interrupt will occur if the CCFn (CCON SFR) and the ECCFn (CCAPMn SFR) bits for the module are both set (see Figure 9).

High Speed Output Mode

In this mode the CEX output (on port 1) associated with the PCA module will toggle each time a match occurs between the PCA counter and the module's capture registers. To activate this mode the TOG, MAT, and ECOM bits in the module's CCAPMn SFR must be set (see Figure 10).

Pulse Width Modulator Mode

All of the PCA modules can be used as PWM outputs. Figure 11 shows the PWM function. The frequency of the output depends on the source for the PCA timer. All of the modules will have the same frequency of output because they all share the PCA timer. The duty cycle of each module is independently variable using the module's capture register CCAPLn. When the value of the PCA CL SFR is less than the value in the module's CCAPLn SFR the output will be low, when it is equal to or greater than the output will be high. When CL overflows from FF to 00, CCAPLn is reloaded with the value in CCAPHn. the allows updating the PWM

without glitches. The PWM and ECOM bits in the module's CCAPMn register must be set to enable the PWM mode.

Enhanced UART

The UART operates in all of the usual modes that are described in the first section of this book for the 80C51. In addition the UART can perform framing error detect by looking for missing stop bits, and automatic address recognition. The 8XC51FB UART also fully supports multiprocessor communication as does the standard 80C51 UART.

When used for framing error detect the UART looks for missing stop bits in the communication. A missing bit will set the FE bit in the SCON register. The FE bit shares the SCON.7 bit with SM0 and the function of SCON.7 is determined by PCON.6 (SMOD0) (see Figure 13). If SMOD0 is set then SCON.7 functions as FE. SCON.7 functions as SM0 when SMOD0 is cleared. When used as FE SCON.7 can only be cleared by software. Refer to Figure 14.

## CMOS single-chip 8-bit microcontrollers 83C51FB/87C51FB

—	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn	MODULE FUNCTION
X	0	0	0	0	0	0	0	No operation
X	X	1	0	0	0	0	X	16-bit capture by a positive-edge trigger on CEXn
X	X	0	1	0	0	0	X	16-bit capture by a negative trigger on CEXn
X	X	1	1	0	0	0	X	16-bit capture by a transition on CEXn
X	1	0	0	1	0	0	X	16-bit Software Timer
X	1	0	0	1	1	0	X	16-bit High Speed Output
X	1	0	0	0	0	1	0	8-bit PWM
X	1	0	0	1	X	0	X	Watchdog Timer

Figure 7. PCA Module Modes (CCAPMn Register)

**Automatic Address Recognition**

Automatic Address Recognition is a feature which allows the UART to recognize certain addresses in the serial bit stream by using hardware to make the comparisons. This feature saves a great deal of software overhead by eliminating the need for the software to examine every serial address which passes by the serial port. This feature is enabled by setting the SM2 bit in SCON. In the 9 bit UART modes, mode 2 and mode 3, the Receive Interrupt flag (RI) will be automatically set when the received byte contains either the "Given" address or the "Broadcast" address. The 9 bit mode requires that the 9th information bit is a 1 to indicate that the received information is an address and not data. Automatic address recognition is shown in Figure 15.

The 8 bit mode is called Mode 1. In this mode the RI flag will be set if SM2 is enabled and the information received has a valid stop bit following the 8 address bits and the information is either a Given or Broadcast address.

Mode 0 is the Shift Register mode and SM2 is ignored.

Using the Automatic Address Recognition feature allows a master to selectively communicate with one or more slaves by invoking the Given slave address or addresses. All of the slaves may be contacted by using the Broadcast address. Two special Function Registers are used to define the slave's address, SADDR, and the address mask, SADEN. SADEN is used to define which bits in the SADDR are to be used and which bits are "don't care". The SADEN mask can be logically ANDed with the

SADDR to create the "Given" address which the master will use for addressing each of the slaves. Use of the Given address allows multiple slaves to be recognized while excluding others. The following examples will help to show the versatility of this scheme:

Slave 0      SADDR = 1100 0000  
              SADEN = 1111 1101  
              Given = 1100 00X0

Slave 1      SADDR = 1100 0000  
              SADEN = 1111 1110  
              Given = 1100 00X0

In the above example SADDR is the same and the SADEN data is used to differentiate between the two slaves. Slave 0 requires a 0 in bit 0 and it ignores bit 1. Slave 1 requires a 0 in bit 1 and bit 0 is ignored. A unique address for Slave 0 would be 1100 0010 since slave 1 requires a 0 in bit 1. A unique address for slave 1 would be 1100 0001 since a 1 in bit 0 will exclude slave 0. Both slaves can be selected at the same time by an address which has bit 0 = 0 (for slave 0) and bit 1 = 0 (for slave 1). Thus, both could be addressed with 1100 0000.

In a more complex system the following could be used to select slaves 1 and 2 while excluding slave 0:

Slave 0      SADDR = 1100 0000  
              SADEN = 1111 1001  
              Given = 1100 0XX0

Slave 1      SADDR = 1110 0000  
              SADEN = 1111 1010  
              Given = 1110 0X0X

Slave 2      SADDR = 1110 0000  
              SADEN = 1111 1100  
              Given = 1110 00XX

In the above example the differentiation among the 3 slaves is in the lower 3 address bits. Slave 0 requires that bit 0 = 0 and it can be uniquely addressed by 1110 0110. Slave 1 requires that bit 1 = 0 and it can be uniquely addressed by 1110 and 0101. Slave 2 requires that bit 2 = 0 and its unique address is 1110 0011. To select Slaves 0 and 1 and exclude Slave 2 use address 1110 0100, since it is necessary to make bit 2 = 1 to exclude slave 2.

The Broadcast Address for each slave is created by taking the logical OR of SADDR and SADEN. Zeros in this result are treated as don't-cares. In most cases, interpreting the don't-cares as ones, the broadcast address will be FF hexadecimal.

Upon reset SADDR (SFR address 0A9H) and SADEN (SFR address 0B9H) are loaded with 0s. This produces a given address of all "don't cares" as well as a Broadcast address of all "don't cares". This effectively disables the Automatic Addressing mode and allows the microcontroller to use standard 80C51 type UART drivers which do not make use of this feature.

**Reduced EMI Mode**

The AO bit (AUXR.0) in the AUXR register when set disables the ALE output.

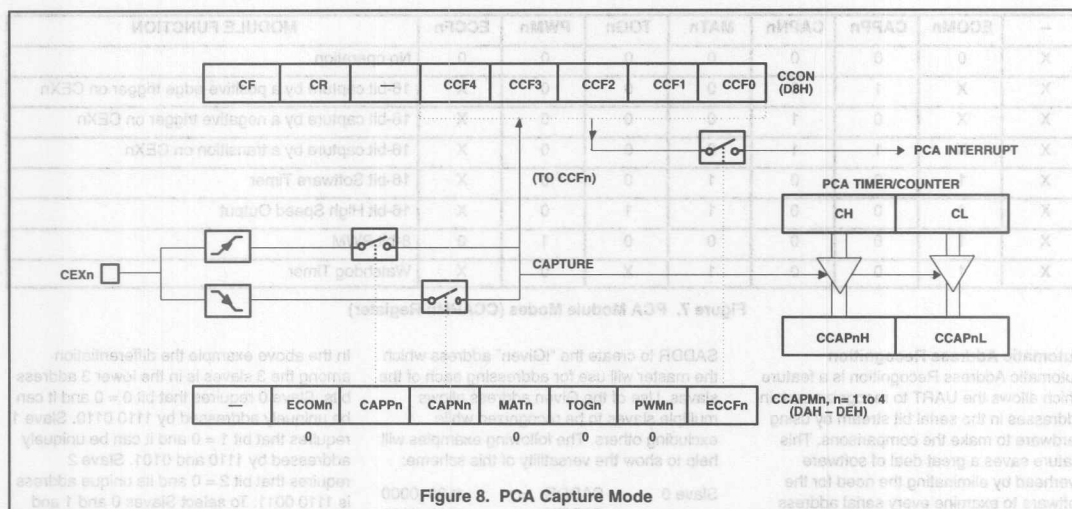
**8XC51FB Reduced EMI Mode****AUXR (0X8E)**

—	—	—	—	—	—	—	—	AO
---	---	---	---	---	---	---	---	----

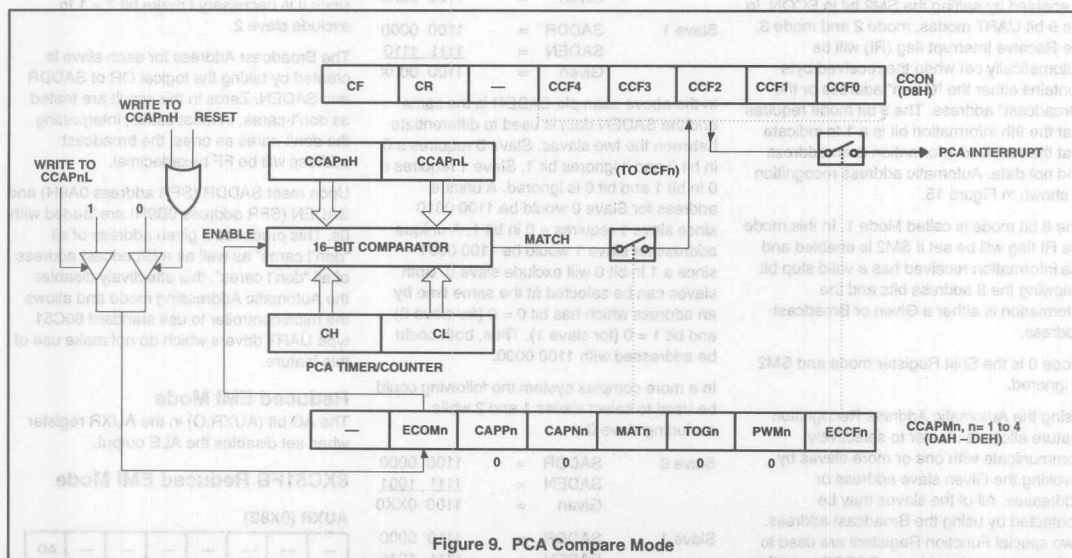
AO: Turns off ALE output.

## CMOS single-chip 8-bit microcontrollers

## 83C51FB/87C51FB



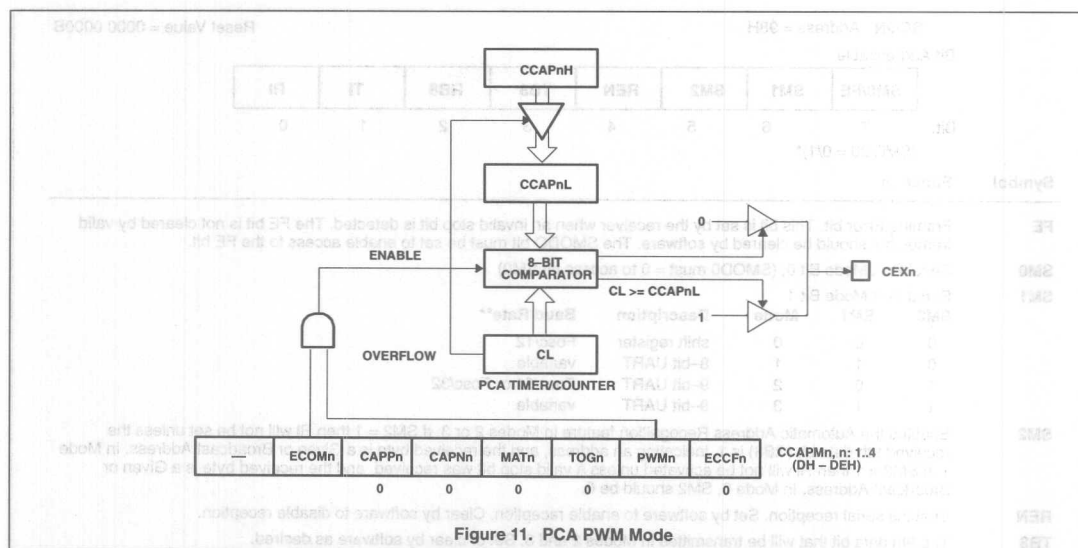
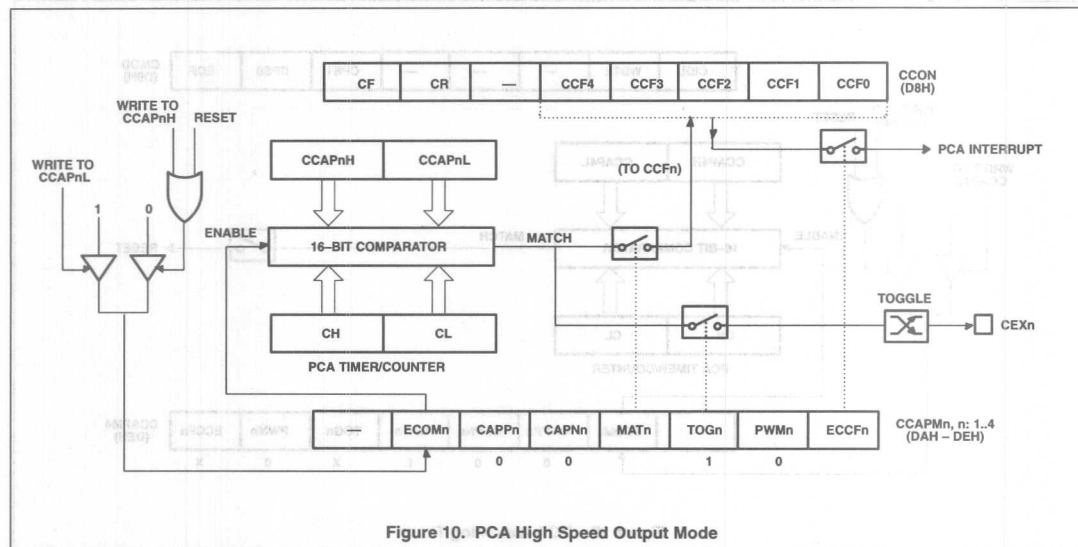
**Figure 8. PCA Capture Mode**



**Figure 9. PCA Compare Mode**

## CMOS single-chip 8-bit microcontrollers

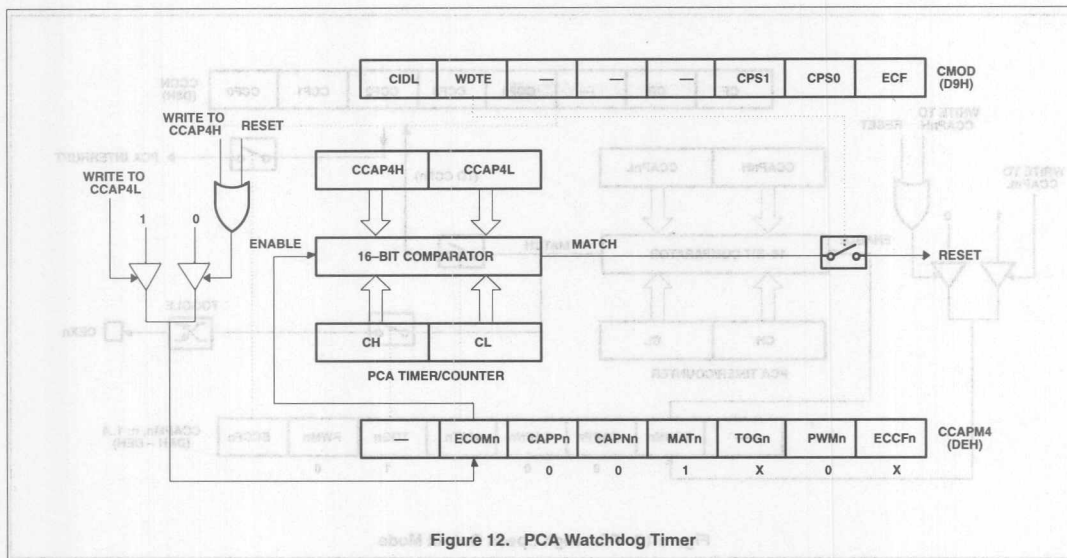
## 83C51FB/87C51FB





## CMOS single-chip 8-bit microcontrollers

## 83C51FB/87C51FB



SCON Address = 98H								Reset Value = 0000 0000B	
Bit Addressable									
	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	
Bit:	7	6	5	4	3	2	1	0	
(SMOD0 = 0/1)*									
Symbol	Function								
FE	Framing Error bit. This bit is set by the receiver when an invalid stop bit is detected. The FE bit is not cleared by valid frames but should be cleared by software. The SMOD0 bit must be set to enable access to the FE bit.								
SM0	Serial Port Mode Bit 0, (SMOD0 must = 0 to access bit SM0)								
SM1	Serial Port Mode Bit 1								
	SM0	SM1	Mode	Description	Baud Rate**				
	0	0	0	shift register	Fosc/12				
	0	1	1	8-bit UART	variable				
	1	0	2	9-bit UART	Fosc/64 or Fosc/32				
	1	1	3	9-bit UART	variable				
SM2	Enables the Automatic Address Recognition feature in Modes 2 or 3. If SM2 = 1 then RI will not be set unless the received 9th data bit (RB8) is 1, indicating an address, and the received byte is a Given or Broadcast Address. In Mode 1, if SM2 = 1 then RI will not be activated unless a valid stop bit was received, and the received byte is a Given or Broadcast Address. In Mode 0, SM2 should be 0.								
REN	Enables serial reception. Set by software to enable reception. Clear by software to disable reception.								
TB8	The 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired.								
RB8	In modes 2 and 3, the 9th data bit that was received. In Mode 1, if SM2 = 0, RB8 is the stop bit that was received. In Mode 0, RB8 is not used.								
TI	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.								
RI	Receive interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM2). Must be cleared by software.								

NOTE:

\*SMOD0 is located at PCON6.

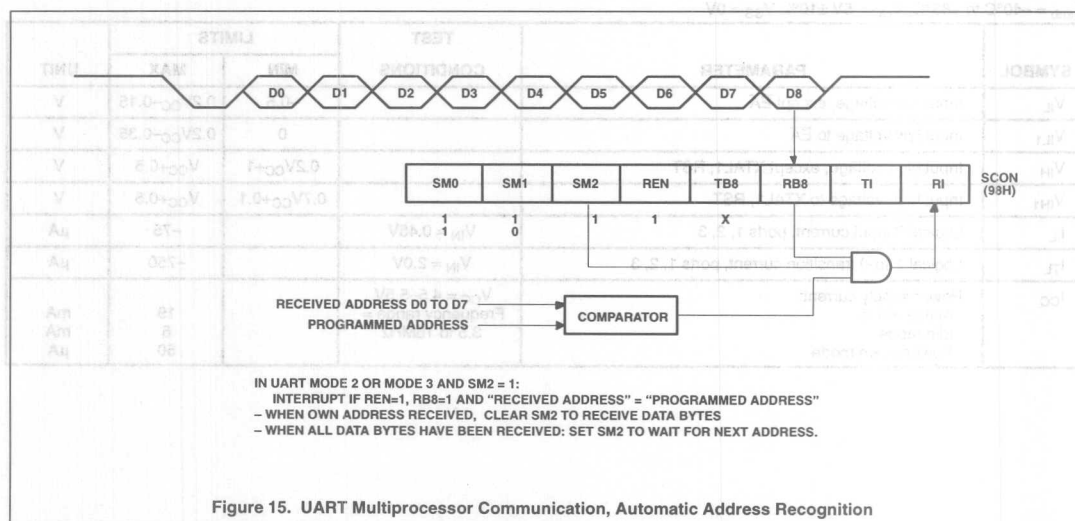
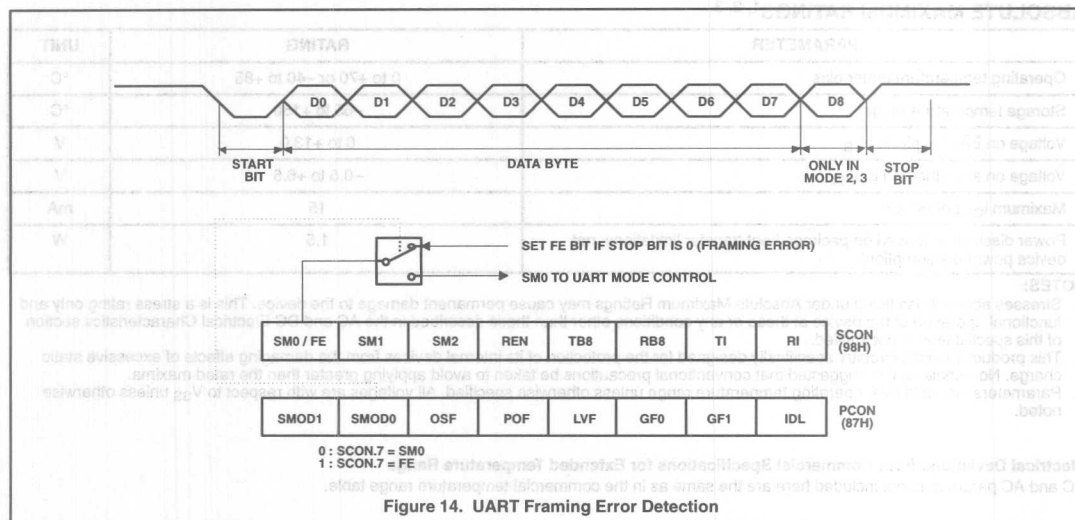
\*\*Fosc = oscillator frequency

Figure 13. SCON: Serial Port Control Register

Figure 13. SCON: Serial Port Control Register

## CMOS single-chip 8-bit microcontrollers

## 83C51FB/87C51FB



CMOS single-chip 8-bit microcontrollers 83C51FB/87C51FB

ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>

PARAMETER	RATING	UNIT
Operating temperature under bias	0 to +70 or -40 to +85	°C
Storage temperature range	-65 to +150	°C
Voltage on $\overline{EA}/V_{PP}$ pin to $V_{SS}$	0 to +13.0	V
Voltage on any other pin to $V_{SS}$	-0.5 to +6.5	V
Maximum $I_{OL}$ per I/O pin	15	mA
Power dissipation (based on package heat transfer limitations, not device power consumption)	1.5	W

NOTES:

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
- This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
- Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to  $V_{SS}$  unless otherwise noted.

Electrical Deviations from Commercial Specifications for Extended Temperature Range

DC and AC parameters not included here are the same as in the commercial temperature range table.

DC ELECTRICAL CHARACTERISTICS

$T_{amb} = -40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			MIN	MAX	
$V_{IL}$	Input low voltage, except $\overline{EA}$		-0.5	$0.2V_{CC}-0.15$	V
$V_{IL1}$	Input low voltage to $\overline{EA}$		0	$0.2V_{CC}-0.35$	V
$V_{IH}$	Input high voltage, except XTAL1, RST		$0.2V_{CC}+1$	$V_{CC}+0.5$	V
$V_{IH1}$	Input high voltage to XTAL1, RST		$0.7V_{CC}+0.1$	$V_{CC}+0.5$	V
$I_{IL}$	Logical 0 input current, ports 1, 2, 3	$V_{IN} = 0.45\text{V}$		-75	$\mu\text{A}$
$I_{TL}$	Logical 1-to-0 transition current, ports 1, 2, 3	$V_{IN} = 2.0\text{V}$		-750	$\mu\text{A}$
$I_{CC}$	Power supply current: Active mode Idle mode Power-down mode	$V_{CC} = 4.5-5.5\text{V}$ , Frequency range = 3.5 to 16MHz		19 6 50	mA mA $\mu\text{A}$

## CMOS single-chip 8-bit microcontrollers 83C51FB/87C51FB

## DC ELECTRICAL CHARACTERISTICS

 $T_{amb} = 0^{\circ}\text{C to } +70^{\circ}\text{C or } -40^{\circ}\text{C to } +85^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$ 

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP <sup>1</sup>	MAX	
$V_{IL}$	Input low voltage, except EA <sup>7</sup>		-0.5		$0.2V_{CC}-0.1$	V
$V_{IL1}$	Input low voltage to EA <sup>7</sup>		0		$0.2V_{CC}-0.3$	V
$V_{IH}$	Input high voltage, except XTAL1, RST <sup>7</sup>		$0.2V_{CC}+0.9$		$V_{CC}+0.5$	V
$V_{IH1}$	Input high voltage, XTAL1, RST <sup>7</sup>		$0.7V_{CC}$		$V_{CC}+0.5$	V
$V_{OL}$	Output low voltage, ports 1, 2, 3 <sup>9</sup>	$I_{OL} = 100\mu\text{A}$ $I_{OL} = 1.6\text{mA}^2$ $I_{OL} = 3.5\text{mA}$			0.3 0.45 1.0	V V V
$V_{OL1}$	Output low voltage, port 0, ALE, PSEN <sup>9</sup>	$I_{OL} = 200\mu\text{A}$ $I_{OL} = 3.2\text{mA}^2$ $I_{OL} = 7.0\text{mA}$			0.3 0.45 1.0	V V V
$V_{OH}$	Output high voltage, ports 1, 2, 3, ALE, PSEN <sup>3</sup>	$I_{OH} = -60\mu\text{A}$ $I_{OH} = -30\mu\text{A}$ $I_{OH} = -10\mu\text{A}$	$V_{CC} - 1.5$ $V_{CC} - 0.7$ $V_{CC} - 0.3$			V V V
$V_{OH1}$	Output high voltage (port 0 in external bus mode), ALE <sup>10</sup> , PSEN <sup>3</sup>	$I_{OH} = -7.0\text{mA}$ $I_{OH} = -3.2\text{mA}$ $I_{OH} = -200\mu\text{A}$	$V_{CC} - 1.5$ $V_{CC} - 0.7$ $V_{CC} - 0.3$			V V V
$I_{IL}$	Logical 0 input current, ports 1, 2, 3 <sup>7</sup>	$V_{IN} = 0.45\text{V}$			-50	$\mu\text{A}$
$I_{TL}$	Logical 1-to-0 transition current, ports 1, 2, 3 <sup>7</sup>	See note 4			-650	$\mu\text{A}$
$I_{LI}$	Input leakage current, port 0	$0.45 V_{IN} < V_{CC} - 0.3$			$\pm 10$	$\mu\text{A}$
$I_{CC}$	Power supply current: <sup>7</sup> Active mode @ 16MHz <sup>5</sup> Idle mode @ 16MHz Power-down mode	See note 6		15 3 10	25 5 50	mA mA $\mu\text{A}$
$R_{RST}$	Internal reset pull-down resistor		50		225	k $\Omega$
$C_{IO}$	Pin capacitance <sup>11</sup> (except EA)				15	pF

## NOTES:

- Typical ratings are not guaranteed. The values listed are at room temperature, 5V.
- Capacitive loading on ports 0 and 2 may cause spurious noise to be superimposed on the  $V_{OL}$ s of ALE and ports 1 and 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100pF), the noise pulse on the ALE pin may exceed 0.8V. In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.  $I_{OL}$  can exceed these conditions provided that no single output sinks more than 5mA and no more than two outputs exceed the test conditions.
- Capacitive loading on ports 0 and 2 may cause the  $V_{OH}$  on ALE and PSEN to momentarily fall below the  $0.9V_{CC}$  specification when the address bits are stabilizing.
- Pins of ports 1, 2 and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when  $V_{IN}$  is approximately 2V.
- $I_{CCMAX}$  at other frequencies is given by: Active mode:  $I_{CCMAX} = 1.50 \times \text{FREQ} + 8$ ; Idle mode:  $I_{CCMAX} = 0.14 \times \text{FREQ} + 2.31$ , where FREQ is the external oscillator frequency in MHz.  $I_{CCMAX}$  is given in mA. See Figure 23.
- See Figures 24 through 27 for  $I_{CC}$  test conditions.
- These values apply only to  $T_{amb} = 0^{\circ}\text{C to } +70^{\circ}\text{C}$ . For  $T_{amb} = -40^{\circ}\text{C to } +85^{\circ}\text{C}$ , see table on previous page.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:  
Maximum  $I_{OL}$  per port pin: 15mA (\*NOTE: This is 85°C specification.)  
Maximum  $I_{OL}$  per 8-bit port: 26mA  
Maximum total  $I_{OL}$  for all outputs: 71mA  
If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
- ALE is tested to  $V_{OH1}$ , except when ALE is off then  $V_{OH}$  is the voltage specification.
- Pin capacitance is less than 25pF. Pin capacitance of ceramic package is less than 15pF (except EA it is 25pF).

## CMOS single-chip 8-bit microcontrollers 83C51FB/87C51FB

## AC ELECTRICAL CHARACTERISTICS

 $T_{amb} = 0^{\circ}\text{C to } +70^{\circ}\text{C or } -40^{\circ}\text{C to } +85^{\circ}\text{C}, V_{CC} = 5\text{V} \pm 10\%, V_{SS} = 0\text{V}^1, 2, 3$ 

SYMBOL	FIGURE	PARAMETER	16MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
$1/t_{CLCL}$	16	Oscillator frequency -4 -5 -A -B			3.5 3.5	16 24	MHz MHz
$t_{LHLL}$	16	ALE pulse width	85		$2t_{CLCL}-40$		ns
$t_{AVLL}$	16	Address valid to ALE low	22		$t_{CLCL}-40$		ns
$t_{LLAX}$	16	Address hold after ALE low	32		$t_{CLCL}-30$		ns
$t_{LLIV}$	16	ALE low to valid instruction in		150	$4t_{CLCL}-100$		ns
$t_{LLPL}$	16	ALE low to PSEN low	32		$t_{CLCL}-30$		ns
$t_{PLPH}$	16	PSEN pulse width	142		$3t_{CLCL}-45$		ns
$t_{PLIV}$	16	PSEN low to valid instruction in		82	$3t_{CLCL}-105$		ns
$t_{PXIX}$	16	Input instruction hold after PSEN	0		0		ns
$t_{PXIZ}$	16	Input instruction float after PSEN		37	$t_{CLCL}-25$		ns
$t_{AVIV}$	16	Address to valid instruction in		207	$5t_{CLCL}-105$		ns
$t_{PLAZ}$	16	PSEN low to address float		10	10		ns
<b>Data Memory</b>							
$t_{RLRH}$	17, 18	RD pulse width	275		$6t_{CLCL}-100$		ns
$t_{WLWH}$	17, 18	WR pulse width	275		$6t_{CLCL}-100$		ns
$t_{RLDV}$	17, 18	RD low to valid data in		147	$5t_{CLCL}-165$		ns
$t_{RHDZ}$	17, 18	Data hold after RD	0		0		ns
$t_{RHDZ}$	17, 18	Data float after RD		65	$2t_{CLCL}-60$		ns
$t_{LLDV}$	17, 18	ALE low to valid data in		350	$8t_{CLCL}-150$		ns
$t_{AVDV}$	17, 18	Address to valid data in		397	$9t_{CLCL}-165$		ns
$t_{LLWL}$	17, 18	ALE low to RD or WR low	137	237	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
$t_{AVWL}$	17, 18	Address valid to WR low or RD low	175		$4t_{CLCL}-75$		ns
$t_{QVWX}$	17, 18	Data valid to WR transition	42		$t_{CLCL}-20$		ns
$t_{WHQX}$	17, 18	Data hold after WR	42		$t_{CLCL}-20$		ns
$t_{QVWH}$	18	Data valid to WR high	287		$7t_{CLCL}-150$		ns
$t_{RLAZ}$	17, 18	RD low to address float		0	0		ns
$t_{WHLH}$	17, 18	RD or WR high to ALE high	40	87	$t_{CLCL}-20$	$t_{CLCL}+25$	ns
<b>External Clock</b>							
$t_{CHCX}$	20	High time	12		20		ns
$t_{CLCX}$	20	Low time	12		20		ns
$t_{CLCH}$	20	Rise time		20		20	ns
$t_{CHCL}$	20	Fall time		20		20	ns
<b>Shift Register</b>							
$t_{XLXL}$	19	Serial port clock cycle time	1		$12t_{CLCL}$		$\mu\text{s}$
$t_{QVXH}$	19	Output data setup to clock rising edge	492		$10t_{CLCL}-133$		ns
$t_{XHGX}$	19	Output data hold after clock rising edge	8		$2t_{CLCL}-117$		ns
$t_{XHDX}$	19	Input data hold after clock rising edge	0		0		ns
$t_{XHDV}$	19	Clock rising edge to input data valid		492	$10t_{CLCL}-133$		ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- Interfacing the 87C51FB to devices with float times up to 45ns is permitted. This limited bus contention will not cause damage to Port 0 drivers.

## CMOS single-chip 8-bit microcontrollers 83C51FB/87C51FB

## EXPLANATION OF THE AC SYMBOLS

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

A – Address  
C – Clock  
D – Input data  
H – Logic level high  
I – Instruction (program memory contents)  
L – Logic level low, or ALE

P – PSEN  
Q – Output data  
R – RD signal  
t – Time  
V – Valid  
W – WR signal  
X – No longer a valid logic level  
Z – Float

Examples:  $t_{AVLL}$  = Time for address valid to ALE low.  
 $t_{LLPL}$  = Time for ALE low to PSEN low.

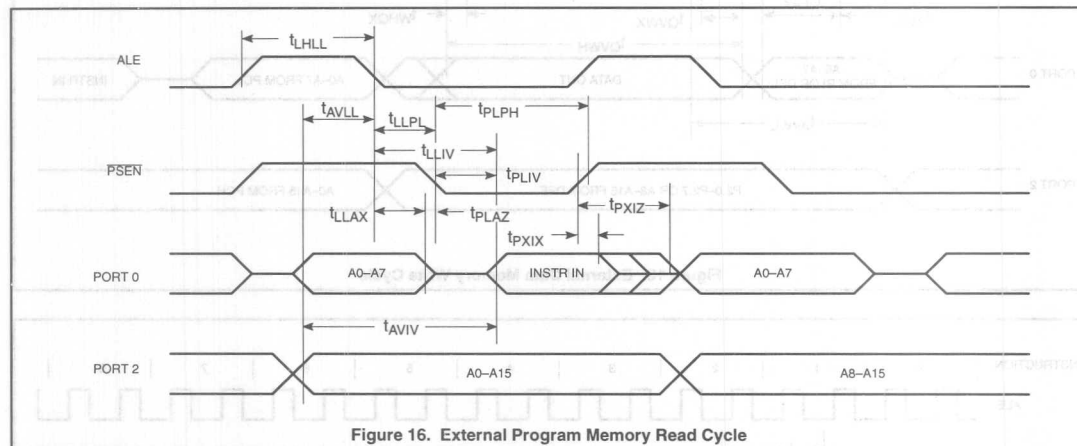


Figure 16. External Program Memory Read Cycle

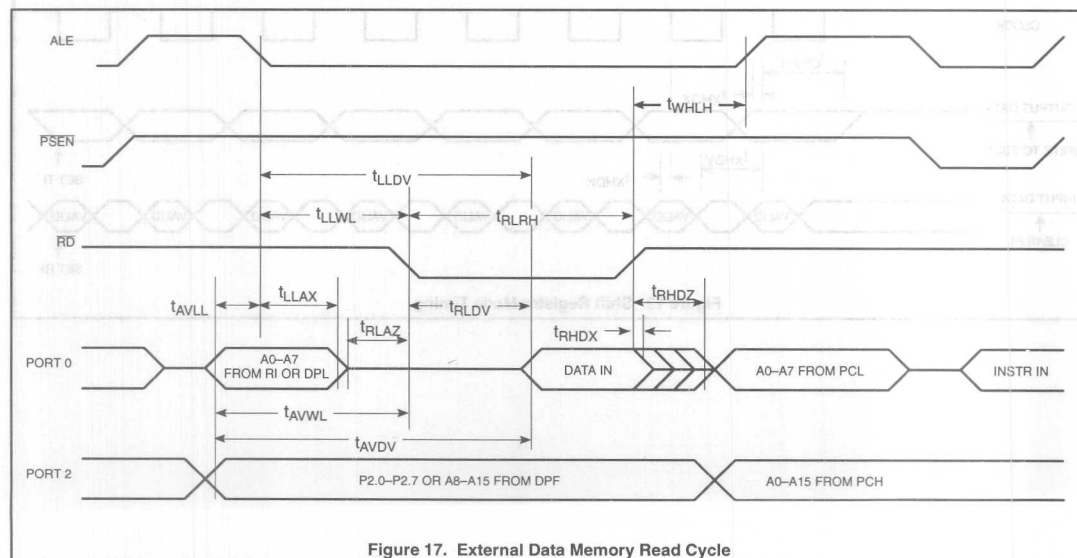


Figure 17. External Data Memory Read Cycle



## CMOS single-chip 8-bit microcontrollers 83C51FB/87C51FB

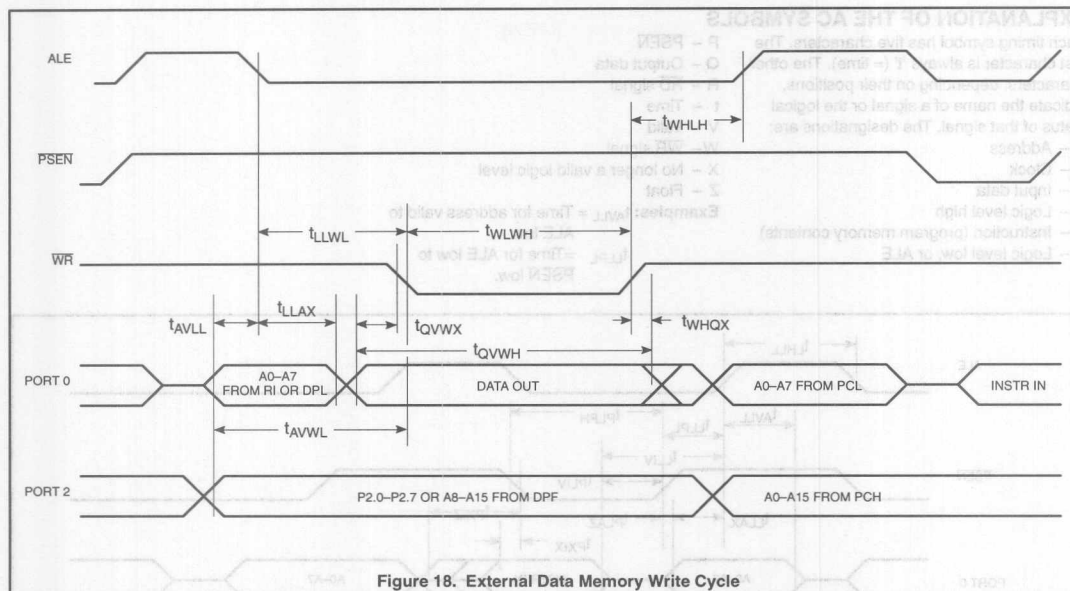


Figure 18. External Data Memory Write Cycle

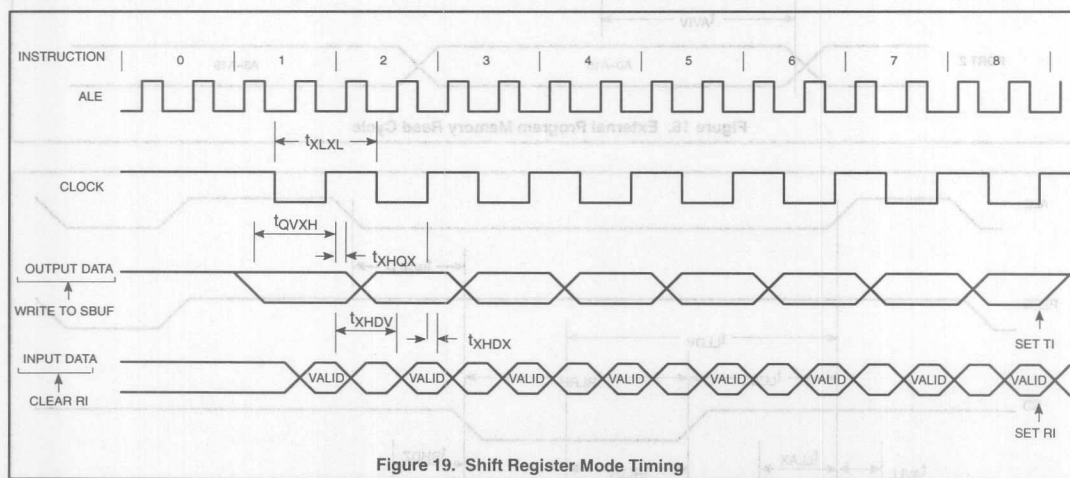
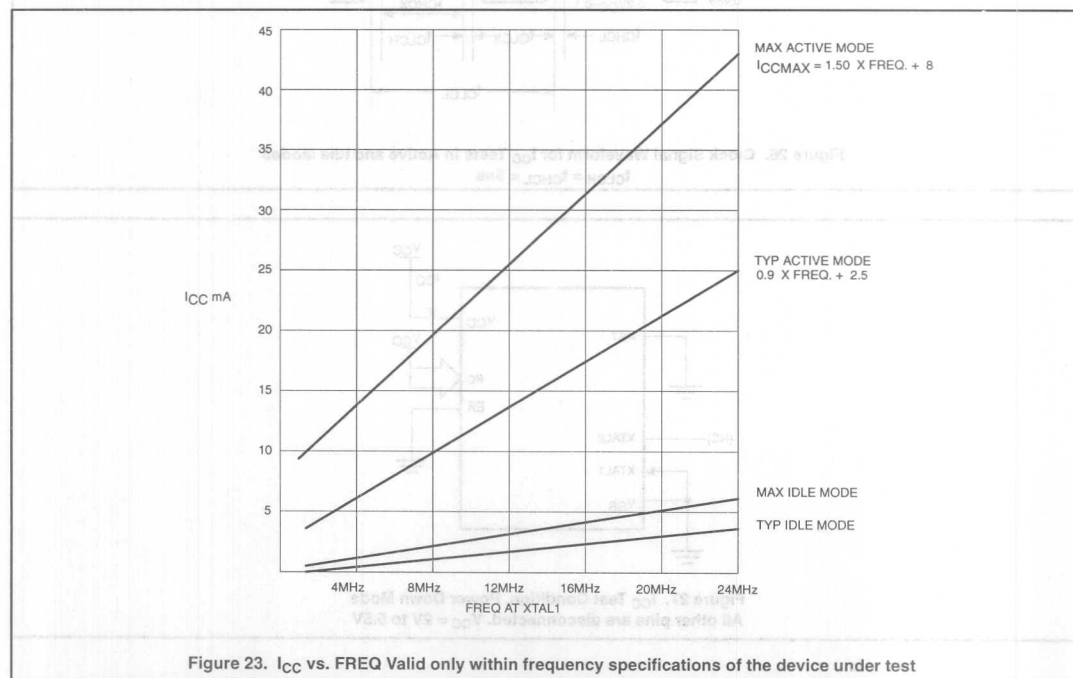
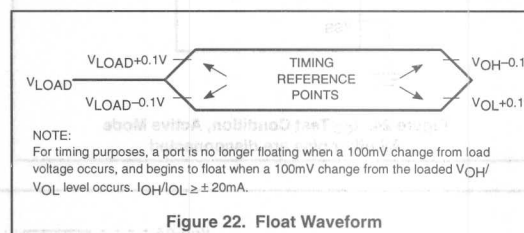
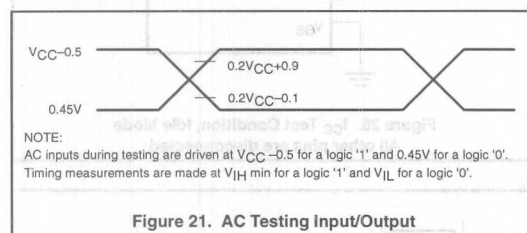
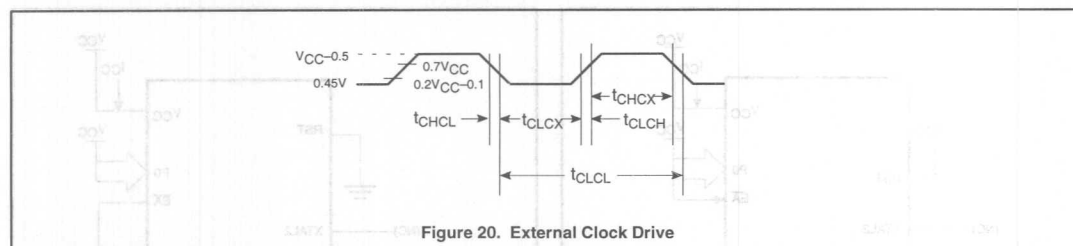
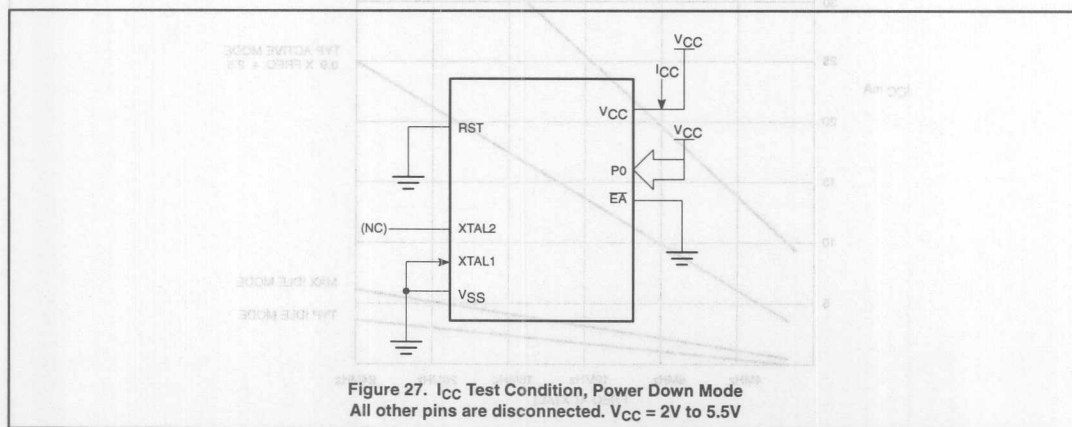
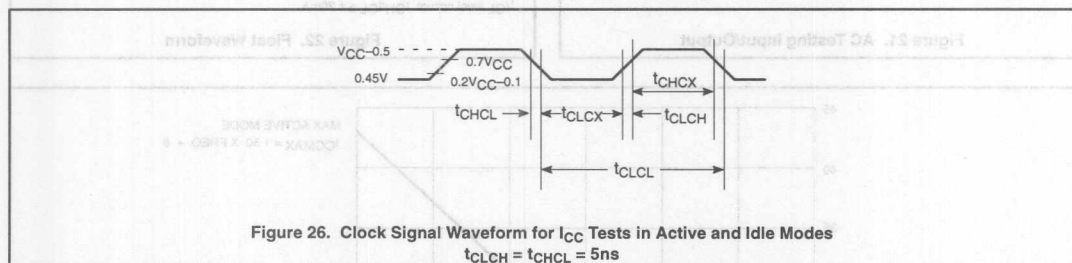
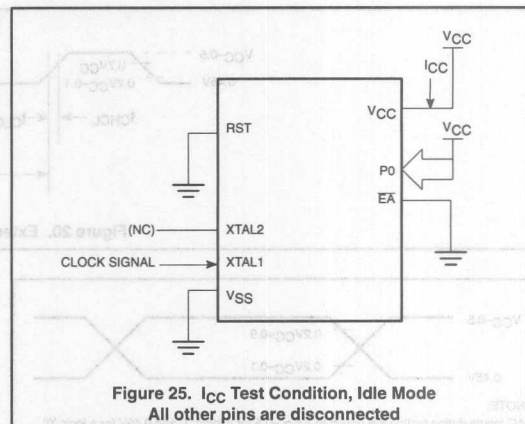
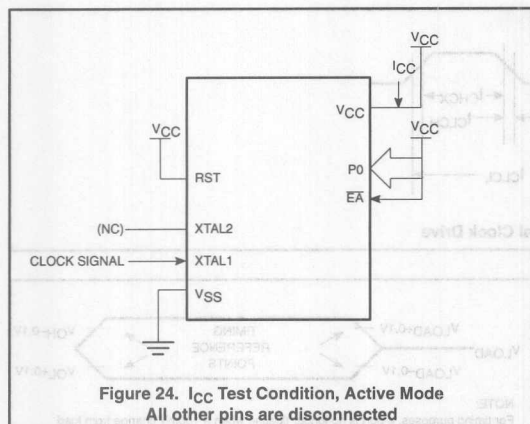


Figure 19. Shift Register Mode Timing

## CMOS single-chip 8-bit microcontrollers 83C51FB/87C51FB



## CMOS single-chip 8-bit microcontrollers 83C51FB/87C51FB



## CMOS single-chip 8-bit microcontrollers 83C51FB/87C51FB

**EPROM CHARACTERISTICS**

The 87C51FB is programmed by using a modified Quick-Pulse Programming™ algorithm. It differs from older methods in the value used for  $V_{PP}$  (programming supply voltage) and in the width and number of the ALE/PROG pulses.

The 87C51FB contains two signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes identify the device as an 87C51FB manufactured by Philips.

Table 3 shows the logic levels for reading the signature byte, and for programming the program memory, the encryption table, and the security bits. The circuit configuration and waveforms for quick-pulse programming are shown in Figures 28 and 29. Figure 30 shows the circuit configuration for normal program memory verification.

**Quick-Pulse Programming**

The setup for microcontroller quick-pulse programming is shown in Figure 28. Note that the 87C51FB is running with a 4 to 6MHz oscillator. The reason the oscillator needs to be running is that the device is executing internal address and program data transfers.

The address of the EPROM location to be programmed is applied to ports 1 and 2, as shown in Figure 28. The code byte to be programmed into that location is applied to port 0. RST, PSEN and pins of ports 2 and 3 specified in Table 3 are held at the 'Program Code Data' levels indicated in Table 3. The ALE/PROG is pulsed low 25 times as shown in Figure 29.

To program the encryption table, repeat the 25 pulse programming sequence for addresses 0 through 1FH, using the 'Pgm Encryption Table' levels. Do not forget that after the encryption table is programmed, verification cycles will produce only encrypted data.

To program the security bits, repeat the 25 pulse programming sequence using the 'Pgm Security Bit' levels. After one security bit is programmed, further programming of the code memory and encryption table is disabled. However, the other security bit can still be programmed.

Note that the EA/ $V_{PP}$  pin must not be allowed to go above the maximum specified  $V_{PP}$  level for any amount of time. Even a narrow glitch above that voltage can cause permanent damage to the device. The  $V_{PP}$  source should be well regulated and free of glitches and overshoot.

**Program Verification**

If security bit 2 has not been programmed, the on-chip program memory can be read out for program verification. The address of the program memory locations to be read is applied to ports 1 and 2 as shown in Figure 30. The other pins are held at the 'Verify Code Data' levels indicated in Table 3. The contents of the address location will be emitted on port 0. External pull-ups are required on port 0 for this operation.

If the encryption table has been programmed, the data presented at port 0 will be the exclusive NOR of the program byte with one of the encryption bytes. The user will have to know the encryption table contents in order to correctly decode the verification data. The encryption table itself cannot be read out.

**Reading the Signature Bytes**

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. The values are:

(030H) = 15H indicates manufactured by Philips

(031H) = B2H indicates 87C51FB

**Program/Verify Algorithms**

Any algorithm in agreement with the conditions listed in Table 3, and which satisfies the timing specifications, is suitable.

**Erase Characteristics**

Erase of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent erasure. **For this and secondary effects, it is recommended that an opaque label be placed over the window.** For elevated temperature or environments where solvents are being used, apply Kapton tape Fluorglas part number 2345-5, or equivalent.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 angstroms) to an integrated dose of at least 15W-s/cm<sup>2</sup>. Exposing the EPROM to an ultraviolet lamp of 12,000μW/cm<sup>2</sup> rating for 20 to 39 minutes, at a distance of about 1 inch, should be sufficient.

Erase leaves the array in an all 1s state.

**Table 3. EPROM Programming Modes**

MODE	RST	PSEN	ALE/PROG	EA/ $V_{PP}$	P2.7	P2.6	P3.7	P3.6
Read signature	1	0	1	1	0	0	0	0
Program code data	1	0	0*	$V_{PP}$	1	0	1	1
Verify code data	1	0	1	1	0	0	1	1
Pgm encryption table	1	0	0*	$V_{PP}$	1	0	1	0
Pgm security bit 1	1	0	0*	$V_{PP}$	1	1	1	1
Pgm security bit 2	1	0	0*	$V_{PP}$	1	1	0	0

**NOTES:**

1. '0' = Valid low for that pin, '1' = valid high for that pin.

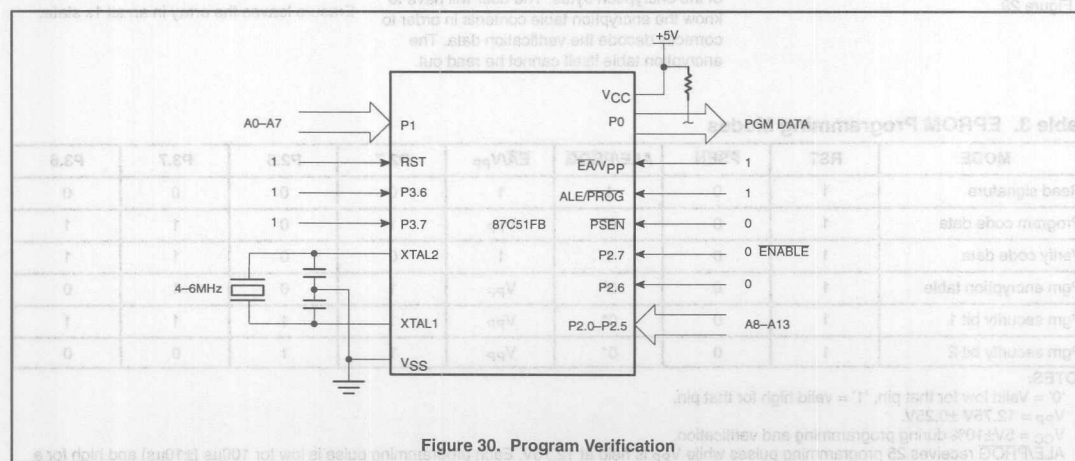
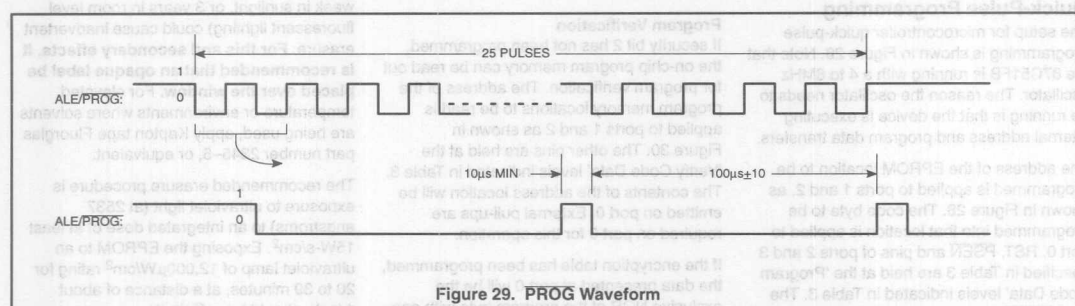
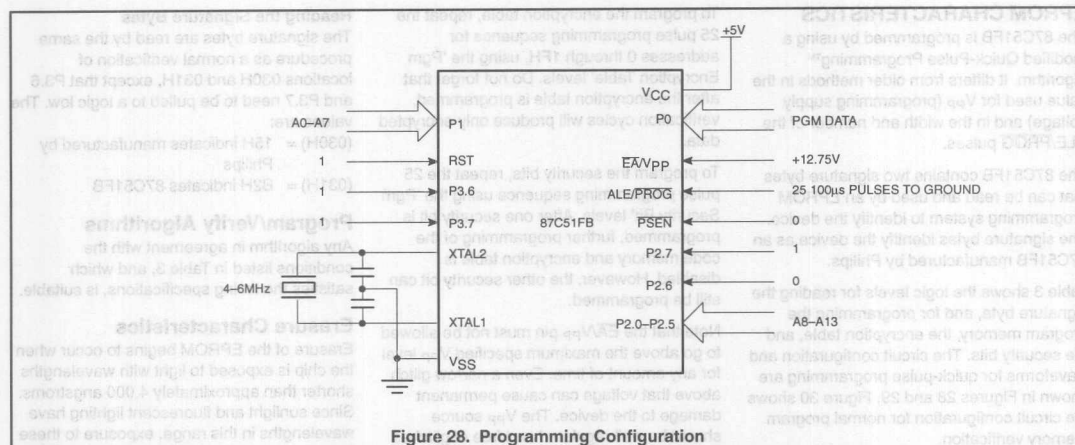
2.  $V_{PP}$  = 12.75V ±0.25V.

3.  $V_{CC}$  = 5V±10% during programming and verification.

\* ALE/PROG receives 25 programming pulses while  $V_{PP}$  is held at 12.75V. Each programming pulse is low for 100μs (±10μs) and high for a minimum of 10μs.

™Trademark phrase of Intel Corporation.

## CMOS single-chip 8-bit microcontrollers 83C51FB/87C51FB

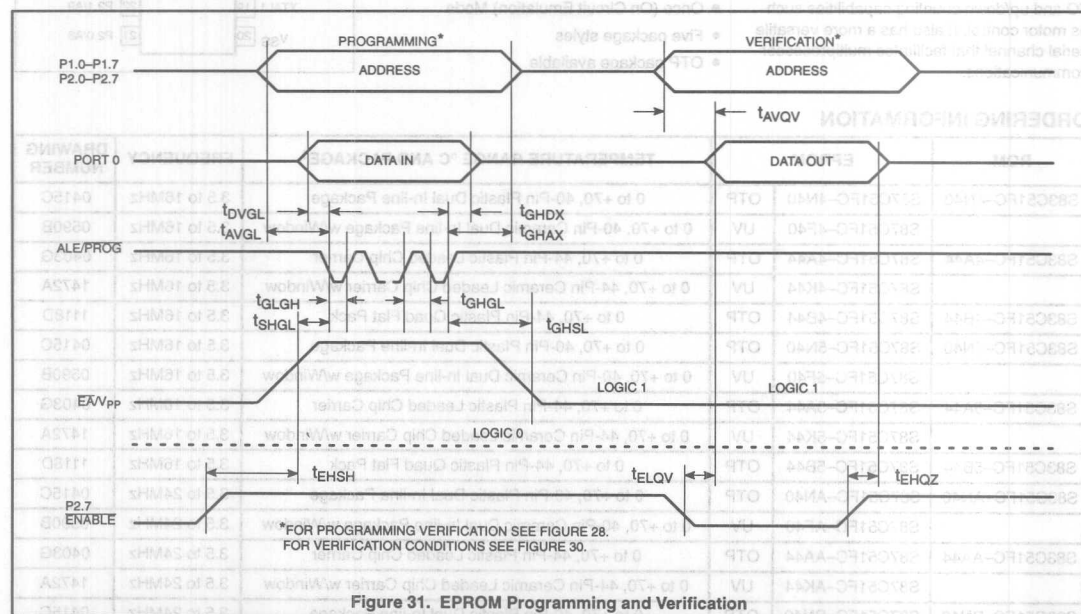


## CMOS single-chip 8-bit microcontrollers 83C51FB/87C51FB

## EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

T<sub>amb</sub> = 21°C to +27°C, V<sub>CC</sub> = 5V±10%, V<sub>SS</sub> = 0V (See Figure 31)

SYMBOL	PARAMETER	MIN	MAX	UNIT
V <sub>PP</sub>	Programming supply voltage	12.5	13.0	V
I <sub>PP</sub>	Programming supply current		50	mA
1/f <sub>CLCL</sub>	Oscillator frequency	4	6	MHz
t <sub>AVGL</sub>	Address setup to PROG low	48t <sub>CLCL</sub>		
t <sub>GHAX</sub>	Address hold after PROG	48t <sub>CLCL</sub>		
t <sub>DVGL</sub>	Data setup to PROG low	48t <sub>CLCL</sub>		
t <sub>GHDX</sub>	Data hold after PROG	48t <sub>CLCL</sub>		
t <sub>ESH</sub>	P2.7 (ENABLE) high to V <sub>PP</sub>	48t <sub>CLCL</sub>		
t <sub>SHGL</sub>	V <sub>PP</sub> setup to PROG low	10		µs
t <sub>GHSL</sub>	V <sub>PP</sub> hold after PROG	10		µs
t <sub>GLGH</sub>	PROG width	90	110	µs
t <sub>AVQV</sub>	Address to data valid		48t <sub>CLCL</sub>	
t <sub>ELQZ</sub>	ENABLE low to data valid		48t <sub>CLCL</sub>	
t <sub>EHQZ</sub>	Data float after ENABLE	0	48t <sub>CLCL</sub>	
t <sub>GHGL</sub>	PROG high to PROG low	10		µs





## DESCRIPTION

The 83C51FC/87C51FC (hereafter collectively called 8XC51FC) Single-Chip 8-Bit Microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 8XC51FC has the same instruction set as the 80C51.

This device provides architectural enhancements that make it applicable in a variety of applications for general control systems. The 87C51FC contains 32k x 8 EPROM memory, the 83C51FC contains 32k x 8 ROM memory, a volatile 256 x 8 read/write data memory, four 8-bit I/O ports, three 16-bit timer/event counters, a Programmable Counter Array (PCA), a multi-source, four-priority-level, nested interrupt structure, an enhanced UART and on-chip oscillator and timing circuits. For systems that require extra capability, the 8XC51FC can be expanded using standard TTL compatible memories and logic.

Its added features make it an even more powerful microcontroller for applications that require pulse width modulation, high-speed I/O and up/down counting capabilities such as motor control. It also has a more versatile serial channel that facilitates multiprocessor communications.

## FEATURES

- 80C51 central processing unit
- 32k x 8 EPROM expandable externally to 64k bytes (87C51FC)
  - Quick Pulse programming algorithm
  - Two level program security system
- 32k x 8 ROM (83C51FC)
- 256 x 8 RAM, expandable externally to 64k bytes
- Three 16-bit timer/counters
  - T2 is an up/down counter
- Programmable Counter Array (PCA)
  - High speed output
  - Capture/compare
  - Pulse Width Modulator
  - Watchdog Timer
- Four 8-bit I/O ports
- Full-duplex enhanced UART
  - Framing error detection
  - Automatic address recognition
- Power control modes
  - Idle mode
  - Power-down mode
- Once (On Circuit Emulation) Mode
- Five package styles
- OTP package available

## PIN CONFIGURATIONS

T2/P1.0	1	40	VCC
T2EX/P1.1	2	39	P0.0/AD0
EC/P1.2	3	38	P0.1/AD1
CEX0/P1.3	4	37	P0.2/AD2
CEX1/P1.4	5	36	P0.3/AD3
CEX2/P1.5	6	35	P0.4/AD4
CEX3/P1.6	7	34	P0.5/AD5
CEX4/P1.7	8	33	P0.6/AD6
RST	9	32	P0.7/AD7
RxD/P3.0	10	31	EA/Vpp
TxD/P3.1	11	30	ALE/PROG
INT0/P3.2	12	29	PSEN
INT1/P3.3	13	28	P2.7/A15
T0/P3.4	14	27	P2.6/A14
T1/P3.5	15	26	P2.5/A13
WR/P3.6	16	25	P2.4/A12
RD/P3.7	17	24	P2.3/A11
XTAL2	18	23	P2.2/A10
XTAL1	19	22	P2.1/A9
VSS	20	21	P2.0/A8

## ORDERING INFORMATION

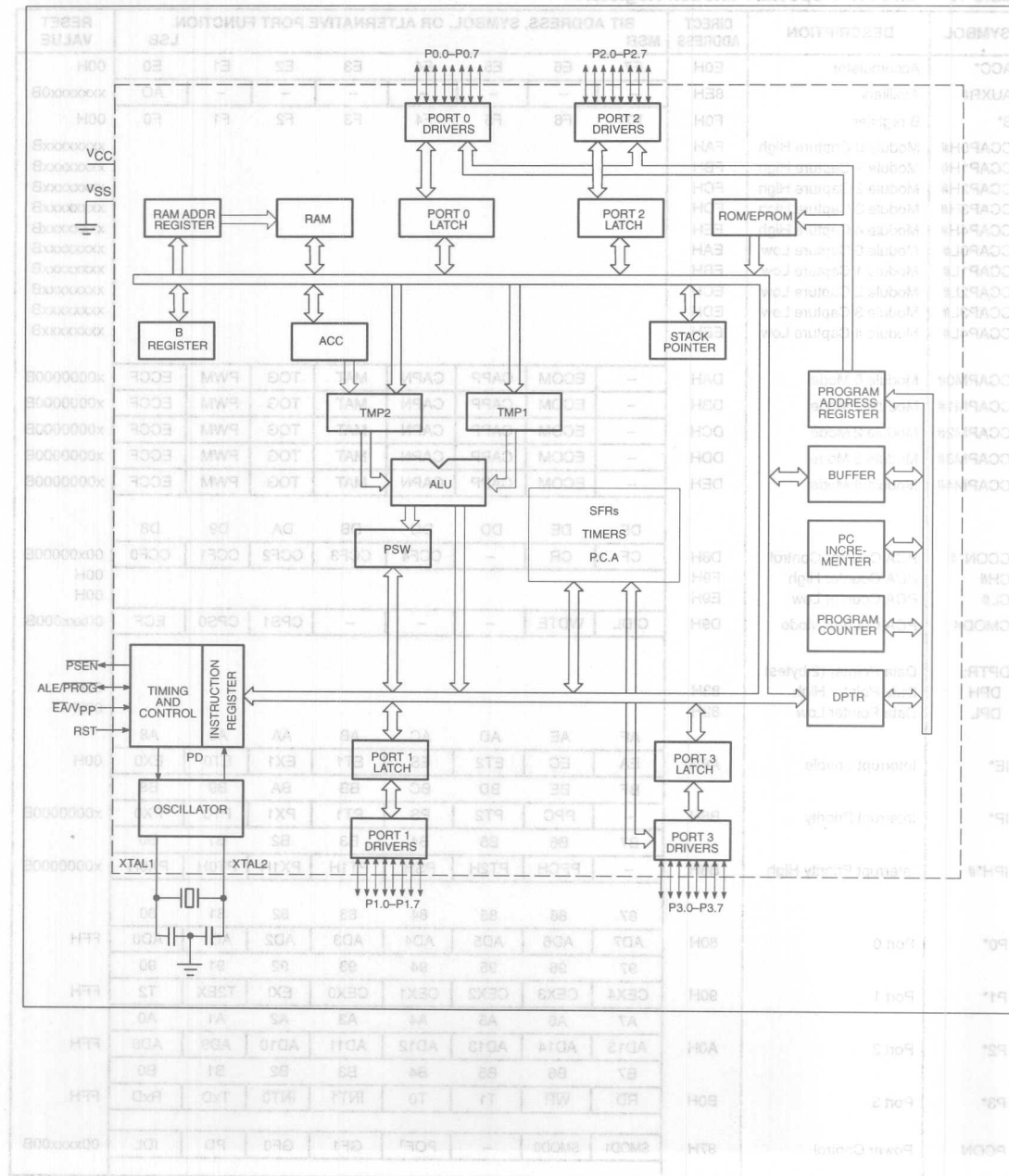
ROM	EPROM		TEMPERATURE RANGE °C AND PACKAGE <sup>1</sup>	FREQUENCY	DRAWING NUMBER
S83C51FC-4N40	S87C51FC-4N40	OTP	0 to +70, 40-Pin Plastic Dual In-line Package	3.5 to 16MHz	0415C
	S87C51FC-4F40	UV	0 to +70, 40-Pin Ceramic Dual In-line Package w/Window	3.5 to 16MHz	0590B
S83C51FC-4A44	S87C51FC-4A44	OTP	0 to +70, 44-Pin Plastic Leaded Chip Carrier	3.5 to 16MHz	0403G
	S87C51FC-4K44	UV	0 to +70, 44-Pin Ceramic Leaded Chip Carrier w/Window	3.5 to 16MHz	1472A
S83C51FC-4B44	S87C51FC-4B44	OTP	0 to +70, 44-Pin Plastic Quad Flat Pack	3.5 to 16MHz	1118D
S83C51FC-5N40	S87C51FC-5N40	OTP	0 to +70, 40-Pin Plastic Dual In-line Package	3.5 to 16MHz	0415C
	S87C51FC-5F40	UV	0 to +70, 40-Pin Ceramic Dual In-line Package w/Window	3.5 to 16MHz	0590B
S83C51FC-5A44	S87C51FC-5A44	OTP	0 to +70, 44-Pin Plastic Leaded Chip Carrier	3.5 to 16MHz	0403G
	S87C51FC-5K44	UV	0 to +70, 44-Pin Ceramic Leaded Chip Carrier w/Window	3.5 to 16MHz	1472A
S83C51FC-5B44	S87C51FC-5B44	OTP	0 to +70, 44-Pin Plastic Quad Flat Pack	3.5 to 16MHz	1118D
S83C51FC-AN40	S87C51FC-AN40	OTP	0 to +70, 40-Pin Plastic Dual In-line Package	3.5 to 24MHz	0415C
	S87C51FC-AF40	UV	0 to +70, 40-Pin Ceramic Dual In-line Package w/Window	3.5 to 24MHz	0590B
S83C51FC-AA44	S87C51FC-AA44	OTP	0 to +70, 44-Pin Plastic Leaded Chip Carrier	3.5 to 24MHz	0403G
	S87C51FC-AK44	UV	0 to +70, 44-Pin Ceramic Leaded Chip Carrier w/Window	3.5 to 24MHz	1472A
S83C51FC-BN40	S87C51FC-BN40	OTP	0 to +70, 40-Pin Plastic Dual In-line Package	3.5 to 24MHz	0415C
	S87C51FC-BF40	UV	0 to +70, 40-Pin Ceramic Dual In-line Package w/Window	3.5 to 24MHz	0590B
S83C51FC-BA44	S87C51FC-BA44	OTP	0 to +70, 44-Pin Plastic Leaded Chip Carrier	3.5 to 24MHz	0403G
	S87C51FC-BK44	UV	0 to +70, 44-Pin Ceramic Leaded Chip Carrier w/Window	3.5 to 24MHz	1472A

### NOTE:

1. OTP = One Time Programmable EPROM. UV = Erasable EPROM.

## CMOS single-chip 8-bit microcontrollers 83C51FC/87C51FC

## BLOCK DIAGRAM



## CMOS single-chip 8-bit microcontrollers 83C51FC/87C51FC

Table 1. 8XC51FC Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB							LSB	
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
AUXR#	Auxiliary	8EH	—	—	—	—	—	—	—	AO	xxxxxxx0B
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
CCAP0H#	Module 0 Capture High	FAH									xxxxxxxB
CCAP1H#	Module 1 Capture High	FBH									xxxxxxxB
CCAP2H#	Module 2 Capture High	FCH									xxxxxxxB
CCAP3H#	Module 3 Capture High	FDH									xxxxxxxB
CCAP4H#	Module 4 Capture High	FEH									xxxxxxxB
CCAP0L#	Module 0 Capture Low	EAH									xxxxxxxB
CCAP1L#	Module 1 Capture Low	EBH									xxxxxxxB
CCAP2L#	Module 2 Capture Low	ECH									xxxxxxxB
CCAP3L#	Module 3 Capture Low	EDH									xxxxxxxB
CCAP4L#	Module 4 Capture Low	EEH									xxxxxxxB
CCAPM0#	Module 0 Mode	DAH	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM1#	Module 1 Mode	DBH	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM2#	Module 2 Mode	DCH	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM3#	Module 3 Mode	DDH	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM4#	Module 4 Mode	DEH	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCON*#	PCA Counter Control	D8H	DF	DE	DD	DC	DB	DA	D9	D8	00x00000B
CH#	PCA Counter High	F9H	CF	CR	—	CCF4	CCF3	CCF2	CCF1	CCF0	00H
CL#	PCA Counter Low	E9H									00H
CMOD#	PCA Counter Mode	D9H	CIDL	WDTE	—	—	—	CPS1	CPS0	ECF	00xxx000B
DPTR:	Data Pointer (2 bytes)										
DPH	Data Pointer High	83H									00H
DPL	Data Pointer Low	82H									00H
IE*	Interrupt Enable	A8H	AF	AE	AD	AC	AB	AA	A9	A8	00H
			EA	EC	ET2	ES	ET1	EX1	ET0	EX0	00H
IP*	Interrupt Priority	B8H	BF	BE	BD	BC	BB	BA	B9	B8	00H
			—	PPC	PT2	PS	PT1	PX1	PT0	PX0	x0000000B
IPH*#	Interrupt Priority High	B8H	B7	B6	B5	B4	B3	B2	B1	B0	00H
			—	PPCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	x0000000B
P0*	Port 0	80H	87	86	85	84	83	82	81	80	FFH
			AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
P1*	Port 1	90H	97	96	95	94	93	92	91	90	FFH
			CEX4	CEX3	CEX2	CEX1	CEX0	EX1	T2EX	T2	FFH
P2*	Port 2	A0H	A7	A6	A5	A4	A3	A2	A1	A0	FFH
			AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8	FFH
P3*	Port 3	B0H	B7	B6	B5	B4	B3	B2	B1	B0	FFH
			RD	WR	T1	T0	INT1	INT0	TxD	RxD	FFH
PCON	Power Control	87H	SMOD1	SMOD0	—	POF1	GF1	GF0	PD	IDL	00xxxx00B

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.

## CMOS single-chip 8-bit microcontrollers 83C51FC/87C51FC

Table 1. 8XC51FC Special Function Registers (Continued)

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB				LSB				
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program Status Word	D0H	CY	AC	F0	RS1	RS0	OV	—	P	00H
RACAP2H#	Timer 2 Capture High	CBH									00H
RACAP2L#	Timer 2 Capture Low	CAH									00H
SADDR#	Slave Address	A9H									00H
SADEN#	Slave Address Mask	B9H									00H
SBUF	Serial Data Buffer	99H									xxxxxxxxxB
			9F	9E	9D	9C	9B	9A	99	98	
SCON*	Serial Control	98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00H
SP	Stack Pointer	81H									07H
			8F	8E	8D	8C	8B	8A	89	88	
TCON*	Timer Control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
			CF	CE	CD	CC	CB	CA	C9	C8	
T2CON*	Timer 2 Control	C8H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	00H
T2MOD#	Timer 2 Mode Control	C9H	—	—	—	—	—	—	—	DCEN	xxxxxxx0B
TH0	Timer High 0	8CH									00H
TH1	Timer High 1	8DH									00H
TH2#	Timer High 2	CDH									00H
TL0	Timer Low 0	8AH									00H
TL1	Timer Low 1	8BH									00H
TL2#	Timer Low 2	CCH									00H
TMOD	Timer Mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H
			C7	C6	C5	C4	C3	C2	C1	C0	

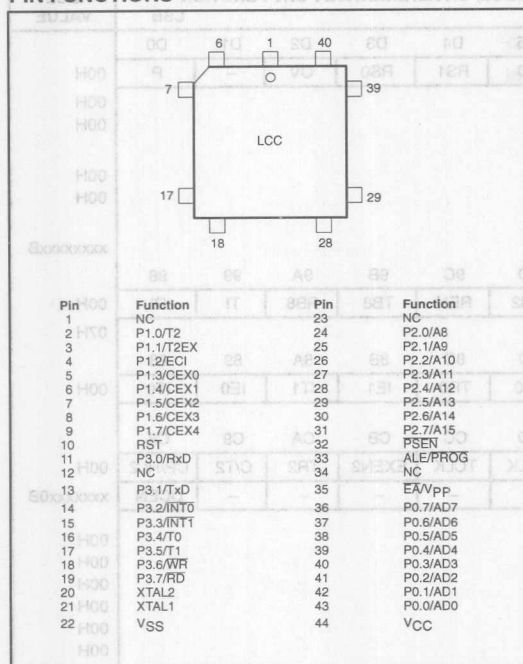
\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.

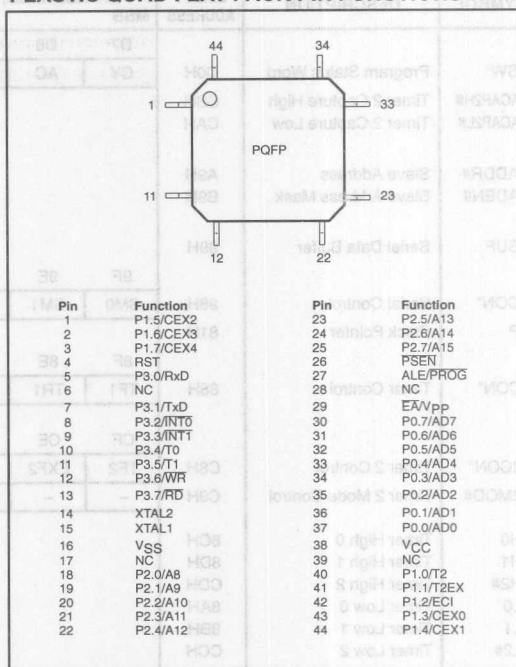
1. Reset value depends on reset source.

NAME AND FUNCTION	TYPE	PORT NUMBER				MEMORIC
		GPIO	GPIO	GPIO	GPIO	
Power Supply: This is the power supply voltage for normal, idle, and power-down operation.	I	18	22	20	24	Vcc
Ground: 0V reference.	I	18	22	20	24	GND
Port 0: Port 0 is an 8-bit bidirectional I/O port. Port 0 pins that have 1A written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed timer/counter and data bus during accesses to external program and data memory in low-power mode. It uses strong internal pull-ups when emitting 1A. Port 0 also outputs the code bytes during program verification and receives code bytes during EPROM programming. External pull-ups are required during program verification.	IO	27-30	27-30	27-30	27-30	P0.0-P0.7
Port 1: Port 1 is an 8-bit bidirectional I/O port with internal pull-ups, except P1.5 and P1.7 which are open-drain. Port 1 pins that have 1A written to them are pulled up by the internal pull-ups and can be used as inputs. As inputs, Port 1 pins that are externally pulled up will source current because of the internal pull-ups (see DC Electrical Characteristics, 11). Port 1 also receives the low-order address bytes during program memory verification. Alternate functions include:	IO	40-41	40-41	40-41	40-41	P1.0-P1.7
TS (P1.0): Timer/Counter 2 external count input/output	I	40	40	40	40	
T2EX (P1.1): Timer/Counter 2 Reload/Direction Control	I	41	41	41	41	
EC0 (P1.2): External Clock Input to the PCA	I	42	42	42	42	
CEX0 (P1.3): Capture/Compare External NO for PCA module 0	IO	43	43	43	43	
CEX1 (P1.4): Capture/Compare External NO for PCA module 1	IO	44	44	44	44	
CEX2 (P1.5): Capture/Compare External NO for PCA module 2	IO	45	45	45	45	
CEX3 (P1.6): Capture/Compare External NO for PCA module 3	IO	46	46	46	46	
CEX4 (P1.7): Capture/Compare External NO for PCA module 4	IO	47	47	47	47	

# CERAMIC AND PLASTIC LEADED CHIP CARRIER PIN FUNCTIONS



# PLASTIC QUAD FLAT PACK PIN FUNCTIONS



## PIN DESCRIPTIONS

MNEMONIC	PIN NUMBER			TYPE	NAME AND FUNCTION
	DIP	LCC	QFP		
VSS	20	22	16	I	Ground: 0V reference.
VCC	40	44	38	I	<b>Power Supply:</b> This is the power supply voltage for normal, idle, and power-down operation.
P0.0–0.7	39–32	43–36	37–30	I/O	<b>Port 0:</b> Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s. Port 0 also outputs the code bytes during program verification and receives code bytes during EPROM programming. External pull-ups are required during program verification.
P1.0–P1.7	1–8	2–9	40–44, 1–3	I/O	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pull-ups, except P1.6 and P1.7 which are open drain. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 1 also receives the low-order address byte during program memory verification. Alternate functions include: <b>T2 (P1.0):</b> Timer/Counter 2 external count input/Clockout <b>T2EX (P1.1):</b> Timer/Counter 2 Reload/Capture/Direction Control <b>ECI (P1.2):</b> External Clock Input to the PCA <b>CEX0 (P1.3):</b> Capture/Compare External I/O for PCA module 0 <b>CEX1 (P1.4):</b> Capture/Compare External I/O for PCA module 1 <b>CEX2 (P1.5):</b> Capture/Compare External I/O for PCA module 2 <b>CEX3 (P1.6):</b> Capture/Compare External I/O for PCA module 3 <b>CEX4 (P1.7):</b> Capture/Compare External I/O for PCA module 4
	1	2	40	I	
	2	3	41	I	
	3	4	42	I	
	4	5	43	I/O	
	5	6	44	I/O	
	6	7	1	I/O	
	7	8	2	I/O	
	8	9	3	I/O	



## CMOS single-chip 8-bit microcontrollers 83C51FC/87C51FC

## PIN DESCRIPTIONS (Continued)

MNEMONIC	PIN NUMBER			TYPE	NAME AND FUNCTION
	DIP	LCC	QFP		
P2.0–P2.7	21–28	24–31	18–25	I/O	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: $I_{IL}$ ). Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register. Some Port 2 pins receive the high order address bits during EPROM programming and verification.
P3.0–P3.7	10–17	11, 13–19	5, 7–13	I/O	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: $I_{IL}$ ). Port 3 also serves the special features of the 80C51 family, as listed below: <b>RxD (P3.0):</b> Serial input port <b>TxD (P3.1):</b> Serial output port <b>INT0 (P3.2):</b> External interrupt <b>INT1 (P3.3):</b> External interrupt <b>T0 (P3.4):</b> Timer 0 external input <b>T1 (P3.5):</b> Timer 1 external input <b>WR (P3.6):</b> External data memory write strobe <b>RD (P3.7):</b> External data memory read strobe
RST	9	10	4	I	<b>Reset:</b> A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to $V_{SS}$ permits a power-on reset using only an external capacitor to $V_{CC}$ .
ALE/PROG	30	33	27	I/O	<b>Address Latch Enable/Program Pulse:</b> Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory. This pin is also the program pulse input (PROG) during EPROM programming.
PSEN	29	32	26	O	<b>Program Store Enable:</b> The read strobe to external program memory. When the 8XC51FC is executing code from the external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory. PSEN is not activated during fetches from internal program memory.
EA/V <sub>PP</sub>	31	35	29	I	<b>External Access Enable/Programming Supply Voltage:</b> EA must be externally held low to enable the device to fetch code from external program memory locations 0000H and 7FFFH. If EA is held high, the device executes from internal program memory unless the program counter contains an address greater than 7FFFH. This pin also receives the 12.75V programming supply voltage ( $V_{PP}$ ) during EPROM programming. If security bit 1 is programmed, EA will be internally latched on Reset.
XTAL1	19	21	15	I	<b>Crystal 1:</b> Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	20	14	O	<b>Crystal 2:</b> Output from the inverting oscillator amplifier.

## NOTE:

To avoid "latch-up" effect at power-on, the voltage on any pin at any time must not be higher than  $V_{CC} + 0.5V$  or  $V_{SS} - 0.5V$ , respectively.

## TIMER 2

This is a 16-bit up or down counter, which can be operated as either a timer or event counter. It can be operated in one of three different modes (autoreload, capture or as the baud rate generator for the UART).

In the autoreload mode the Timer can be set to count up or down by setting or clearing the bit DCEN in the T2CON Special Function Register. The SFR's RCAP2H and RCAP2L are used to reload the Timer upon overflow or a 1-to-0 transition on the T2EX input (P1.1).

In the Capture mode Timer 2 can either set TF2 and generate an interrupt or capture its value. To capture Timer 2 in response to a 1-to-0 transition on the T2EX input, the EXEN2 bit in the T2CON must be set. Timer 2 is then captured in SFR's RCAP2H and RCAP2L.

As the baud rate generator, Timer 2 is selected by setting TCLK and/or RCLK in T2CON. As the baud rate generator Timer 2 is incremented at  $1/2$  the oscillator frequency.

## POWER OFF FLAG

The Power Off Flag (POF) is set by on-chip circuitry when the  $V_{CC}$  level on the 8XC51FC rises from 0 to 5V. The POF bit can be set or cleared by software allowing a user to determine if the reset is the result of a power-on or a warm start after powerdown. The  $V_{CC}$  level must remain above 3V for the POF to remain unaffected by the  $V_{CC}$  level.



## CMOS single-chip 8-bit microcontrollers 83C51FC/87C51FC

## OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

### Reset

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. To insure a good power-on reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few milliseconds) plus two machine cycles. At power-on, the voltage on V<sub>CC</sub> and RST must come up at the same time for a proper start-up.

### Idle Mode

In the idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

### Power-Down Mode

To save even more power, a Power Down mode can be invoked by software. In this

mode, the oscillator is stopped and the instruction that invoked Power Down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power Down mode is terminated.

On the 8XC51FC either a hardware reset or external interrupt can use an exit from Power Down. Reset redefines all the SFRs but does not change the on-chip RAM. An external interrupt allows both the SFRs and the on-chip RAM to retain their values.

To properly terminate Power Down the reset or external interrupt should not be executed before V<sub>CC</sub> is restored to its normal operating level and must be held active long enough for the oscillator to restart and stabilize (normally less than 10ms).

With an external interrupt, INT0 and INT1 must be enabled and configured as level-sensitive. Holding the pin low restarts the oscillator but bringing the pin back high completes the exit. Once the interrupt is serviced, the next instruction to be executed after RETI will be the one following the instruction that put the device into Power Down.

### Design Consideration

- When the idle mode is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

### ONCE™ Mode

The ONCE ("On-Circuit Emulation") Mode facilitates testing and debugging of systems

using the 8XC51FC without the 8XC51FC having to be removed from the circuit. The ONCE Mode is invoked by:

1. Pull ALE low while the device is in reset and PSEN is high;
2. Hold ALE low as RST is deactivated.

While the device is in ONCE Mode, the Port 0 pins go into a float state, and the other port pins and ALE and PSEN are weakly pulled high. The oscillator circuit remains active. While the 8XC51FC is in this mode, an emulator or test CPU can be used to drive the circuit. Normal operation is restored when a normal reset is applied.

### Programmable Clock-Out

The 8XC51FC has a new feature. A 50% duty cycle clock can be programmed to come out on P1.0. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed (1) to input the external clock for Timer/Counter 2 or (2) to output a 50% duty cycle clock ranging from 61Hz to 4MHz at a 16MHz operating frequency.

To configure the Timer/Counter 2 as a clock generator, bit C/T2 (in T2CON) must be cleared and bit T2OE in T2MOD must be set. Bit TR2 (T2CON.2) also must be set to start the timer.

The Clock-Out frequency depends on the oscillator frequency and the reload value of Timer 2 capture registers (RCAP2H, RCAP2L) as shown in this equation:

$$\text{Oscillator Frequency} \\ 4 \times (65536 - \text{RCAP2H, RCAP2L})$$

In the Clock-Out mode Timer 2 roll-overs will not generate an interrupt. This is similar to when it is used as a baud-rate generator. It is possible to use Timer 2 as a baud-rate generator and a clock generator simultaneously. Note, however, that the baud-rate and the Clock-Out frequency will be the same.

Table 2. External Pin Status During Idle and Power-Down Mode

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

## CMOS single-chip 8-bit microcontrollers 83C51FC/87C51FC

**Programmable Counter Array (PCA)**

The Programmable Counter Array is a special Timer that has five 16-bit capture/compare modules associated with it. Each of the modules can be programmed to operate in one of four modes: rising and/or falling edge capture, software timer, high-speed output, or pulse width modulator. Each module has a pin associated with it in port 1. Module 0 is connected to P1.3(CEX0), module 1 to P1.4(CEX1), etc. The basic PCA configuration is shown in Figure 1.

The PCA timer is a common time base for all five modules and can be programmed to run at: 1/12 the oscillator frequency, 1/4 the oscillator frequency, the Timer 0 overflow, or the input on the ECI pin (P1.2). The timer count source is determined from the CPS1 and CPS0 bits in the CMOD SFR as follows (see Figure 4):

CPS1	CPS0	PCA Timer Count Source
0	0	1/12 oscillator frequency
0	1	1/4 oscillator frequency
1	0	Timer 0 overflow
1	1	External Input at ECI pin

In the CMOD SFR are three additional bits associated with the PCA. They are CIDL which allows the PCA to stop during idle mode, WDTE which enables or disables the watchdog function on module 4, and ECF

which when set causes an interrupt and the PCA overflow flag CF (in the CCON SFR) to be set when the PCA timer overflows. These functions are shown in Figure 2.

The watchdog timer function is implemented in module 4 (see Figure 12).

The CCON SFR contains the run control bit for the PCA and the flags for the PCA timer (CF) and each module (refer to Figure 5). To run the PCA the CR bit (CCON.6) must be set by software. The PCA is shut off by clearing this bit. The CF bit (CCON.7) is set when the PCA counter overflows and an interrupt will be generated if the ECF bit in the CMOD register is set. The CF bit can only be cleared by software. Bits 0 through 4 of the CCON register are the flags for the modules (bit 0 for module 0, bit 1 for module 1, etc.) and are set by hardware when either a match or a capture occurs. These flags also can only be cleared by software. The PCA interrupt system shown in Figure 3.

Each module in the PCA has a special function register associated with it. These registers are: CCAPM0 for module 0, CCAPM1 for module 1, etc. (see Figure 6). The registers contain the bits that control the mode that each module will operate in. The ECCF bit (CCAPMn.0 where n=0, 1, 2, 3, or 4 depending on the module) enables the CCF flag in the CCON SFR to generate an

interrupt when a match or compare occurs in the associated module. PWM (CCAPMn.1) enables the pulse width modulation mode.

The TOG bit (CCAPMn.2) when set causes the CEX output associated with the module to toggle when there is a match between the PCA counter and the module's capture/compare register. The match bit MAT (CCAPMn.3) when set will cause the CCFn bit in the CCON register to be set when there is a match between the PCA counter and the module's capture/compare register.

The next two bits CAPN (CCAPMn.4) and CAPP (CCAPMn.5) determine the edge that a capture input will be active on. The CAPN bit enables the negative edge, and the CAPP bit enables the positive edge. If both bits are set both edges will be enabled and a capture will occur for either transition. The last bit in the register ECOM (CCAPMn.6) when set enables the comparator function. Figure 7 shows the CCAPMn settings for the various PCA functions.

There are two additional registers associated with each of the PCA modules. They are CCAPnH and CCAPnL and these are the registers that store the 16-bit count when a capture occurs or a compare should occur. When a module is used in the PWM mode these registers are used to control the duty cycle of the output.

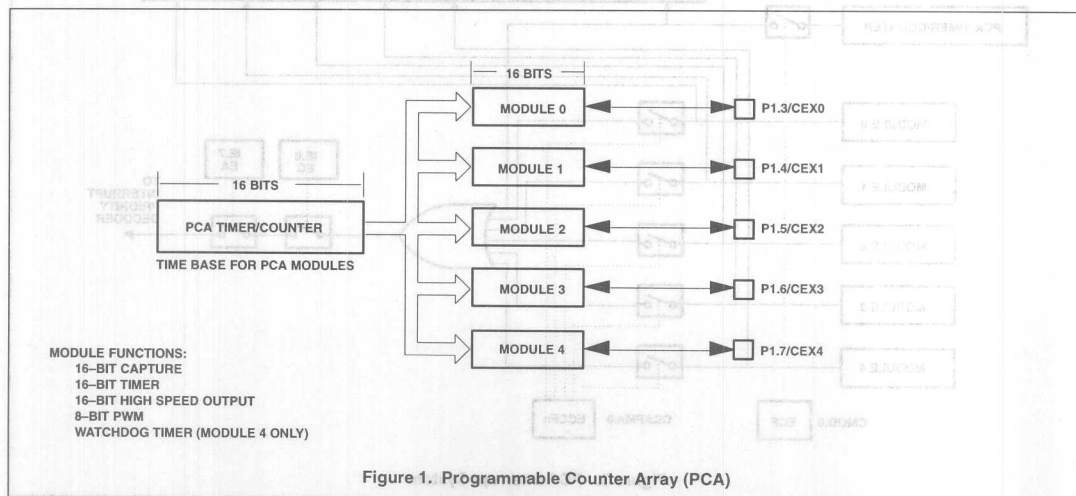
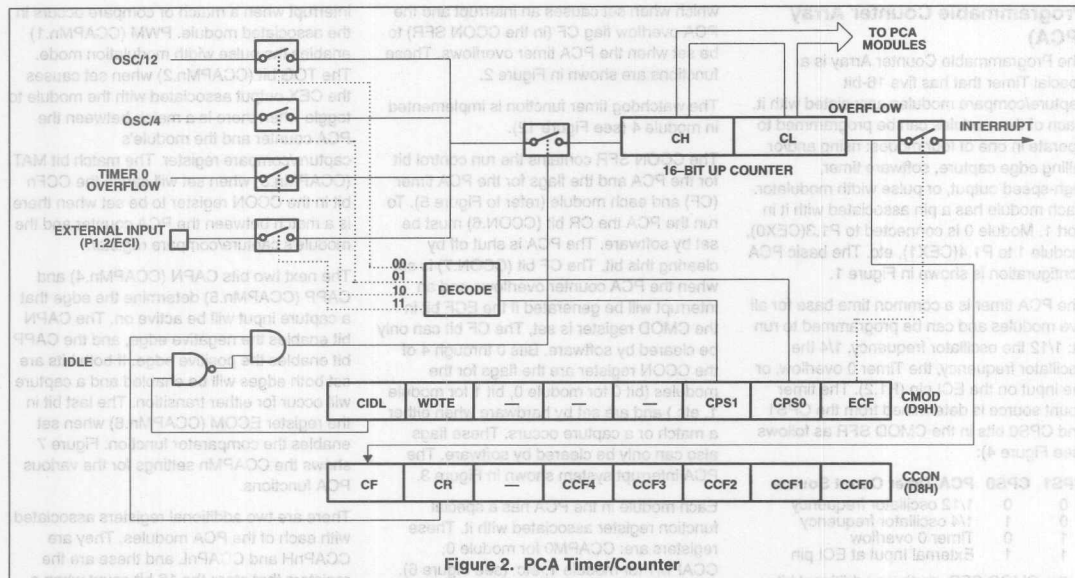
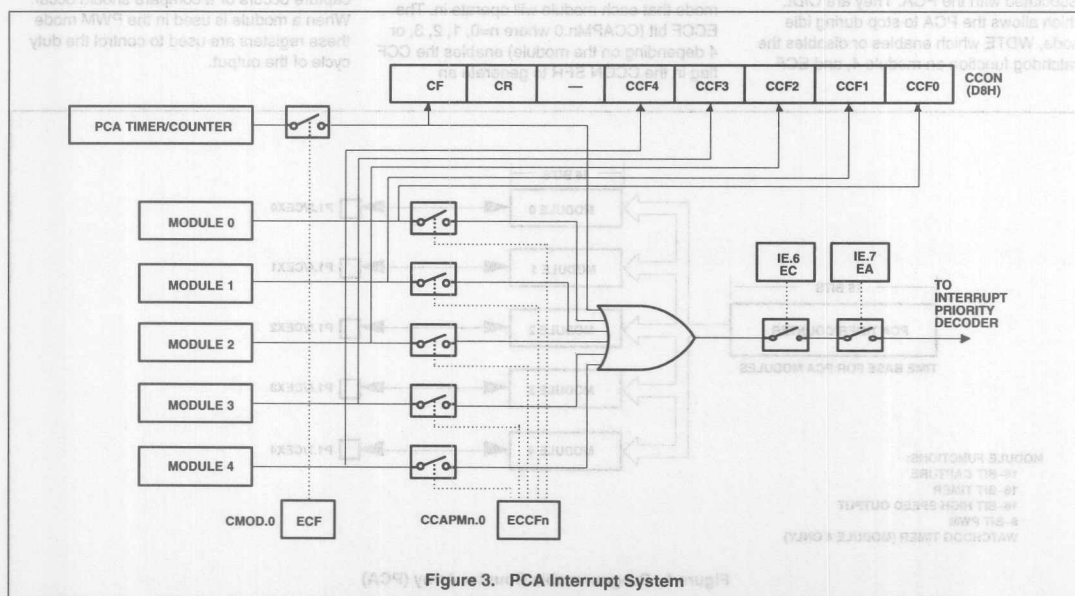


Figure 1. Programmable Counter Array (PCA)

## CMOS single-chip 8-bit microcontrollers 83C51FC/87C51FC

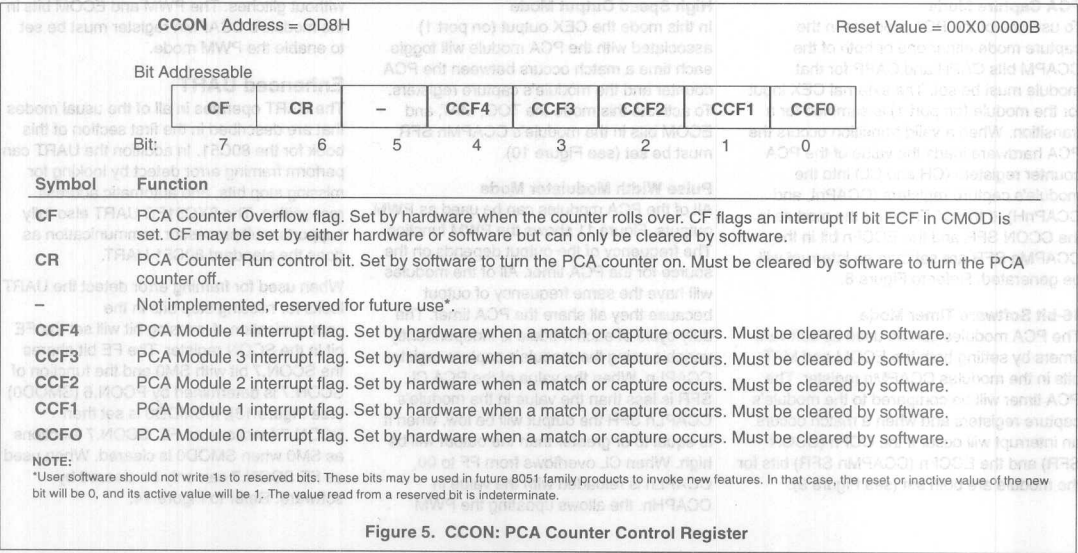
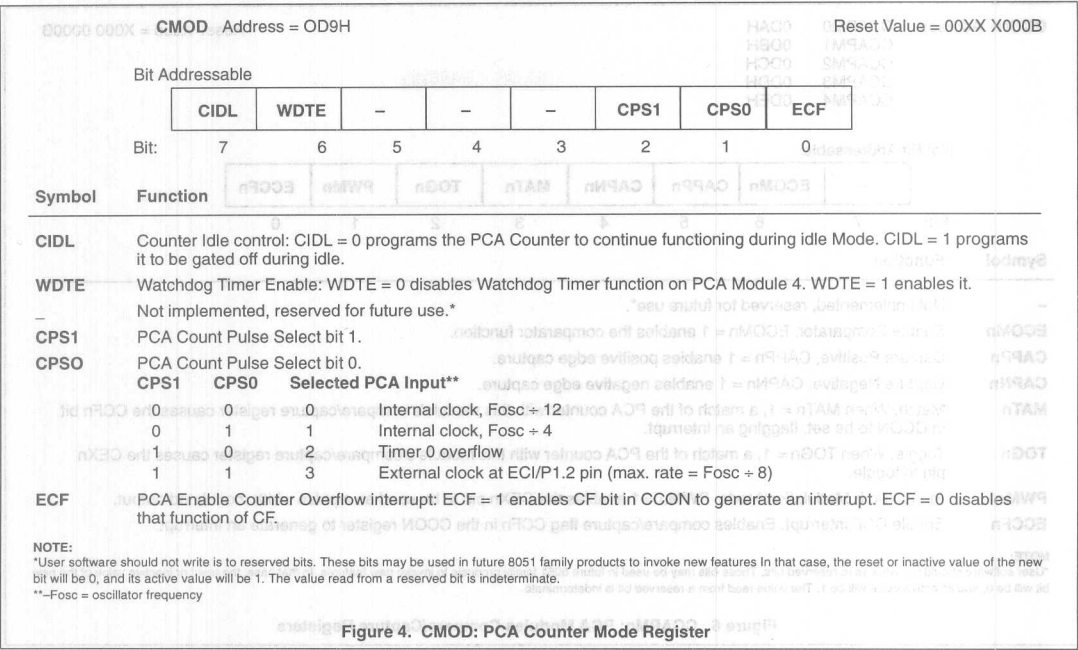


**Figure 2. PCA Timer/Counter**



**Figure 3. PCA Interrupt System**

CMOS single-chip 8-bit microcontrollers 83C51FC/87C51FC



## CMOS single-chip 8-bit microcontrollers 83C51FC/87C51FC

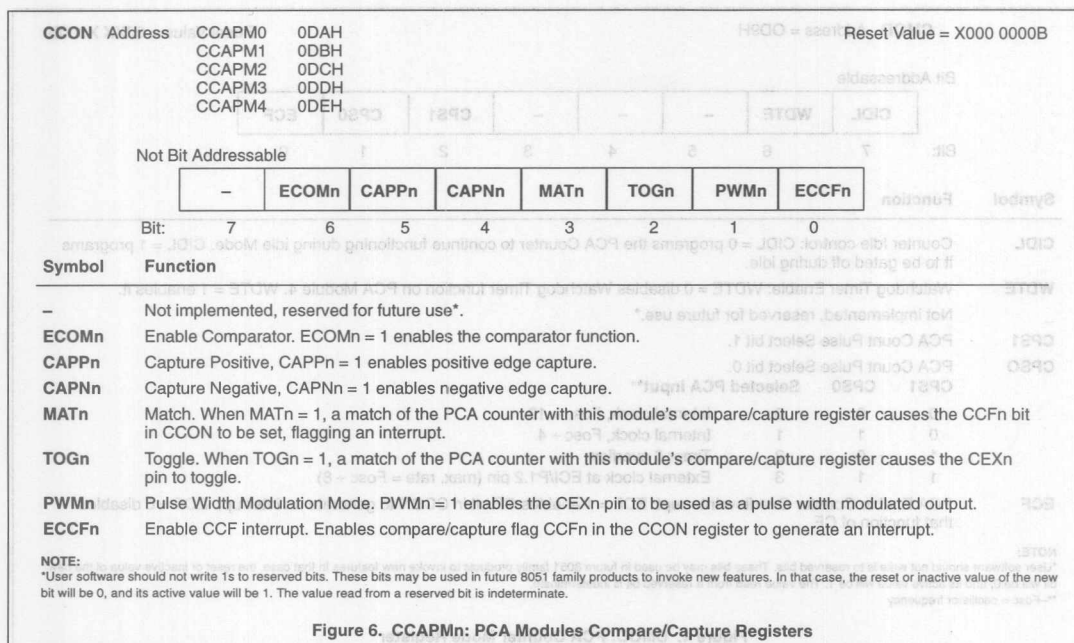


Figure 6. CCAPMn: PCA Modules Compare/Capture Registers

**PCA Capture Mode**

To use one of the PCA modules in the capture mode either one or both of the CCAPM bits CAPN and CAPP for that module must be set. The external CEX input for the module (on port 1) is sampled for a transition. When a valid transition occurs the PCA hardware loads the value of the PCA counter registers (CH and CL) into the module's capture registers (CCAPnL and CCAPnH). If the CCFn bit for the module in the CCON SFR and the ECCFn bit in the CCAPMn SFR are set then an interrupt will be generated. Refer to Figure 8.

**16-bit Software Timer Mode**

The PCA modules can be used as software timers by setting both the ECOM and MAT bits in the modules CCAPMn register. The PCA timer will be compared to the module's capture registers and when a match occurs an interrupt will occur if the CCFn (CCON SFR) and the ECCFn (CCAPMn SFR) bits for the module are both set (see Figure 9).

**High Speed Output Mode**

In this mode the CEX output (on port 1) associated with the PCA module will toggle each time a match occurs between the PCA counter and the module's capture registers. To activate this mode the TOG, MAT, and ECOM bits in the module's CCAPMn SFR must be set (see Figure 10).

**Pulse Width Modulator Mode**

All of the PCA modules can be used as PWM outputs. Figure 11 shows the PWM function. The frequency of the output depends on the source for the PCA timer. All of the modules will have the same frequency of output because they all share the PCA timer. The duty cycle of each module is independently variable using the module's capture register CCAPL<sub>n</sub>. When the value of the PCA CL SFR is less than the value in the module's CCAPL<sub>n</sub> SFR the output will be low, when it is equal to or greater than the output will be high. When CL overflows from FF to 00, CCAPL<sub>n</sub> is reloaded with the value in CCAPH<sub>n</sub>. the allows updating the PWM

without glitches. The PWM and ECOM bits in the module's CCAPMn register must be set to enable the PWM mode.

**Enhanced UART**

The UART operates in all of the usual modes that are described in the first section of this book for the 80C51. In addition the UART can perform framing error detect by looking for missing stop bits, and automatic address recognition. The 8XC51FC UART also fully supports multiprocessor communication as does the standard 80C51 UART.

When used for framing error detect the UART looks for missing stop bits in the communication. A missing bit will set the FE bit in the SCON register. The FE bit shares the SCON.7 bit with SM0 and the function of SCON.7 is determined by PCON.6 (SMOD0) (see Figure 13). If SMOD0 is set then SCON.7 functions as FE. SCON.7 functions as SM0 when SMOD0 is cleared. When used as FE SCON.7 can only be cleared by software. Refer to Figure 14.



## CMOS single-chip 8-bit microcontrollers 83C51FC/87C51FC

—	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn	MODULE FUNCTION
X	0	0	0	0	0	0	0	No operation
X	X	0	0	0	0	0	X	16-bit capture by a positive-edge trigger on CEXn
X	X	0	1	0	0	0	X	16-bit capture by a negative trigger on CEXn
X	X	1	1	0	0	0	X	16-bit capture by a transition on CEXn
X	1	0	0	1	0	0	X	16-bit Software Timer
X	1	0	0	1	1	0	X	16-bit High Speed Output
X	1	0	0	0	0	1	0	8-bit PWM
X	1	0	0	1	X	0	X	Watchdog Timer

Figure 7. PCA Module Modes (CCAPMn Register)

**Automatic Address Recognition**

Automatic Address Recognition is a feature which allows the UART to recognize certain addresses in the serial bit stream by using hardware to make the comparisons. This feature saves a great deal of software overhead by eliminating the need for the software to examine every serial address which passes by the serial port. This feature is enabled by setting the SM2 bit in SCON. In the 9 bit UART modes, mode 2 and mode 3, the Receive Interrupt flag (RI) will be automatically set when the received byte contains either the "Given" address or the "Broadcast" address. The 9 bit mode requires that the 9th information bit is a 1 to indicate that the received information is an address and not data. Automatic address recognition is shown in Figure 15.

The 8 bit mode is called Mode 1. In this mode the RI flag will be set if SM2 is enabled and the information received has a valid stop bit following the 8 address bits and the information is either a Given or Broadcast address.

Mode 0 is the Shift Register mode and SM2 is ignored.

Using the Automatic Address Recognition feature allows a master to selectively communicate with one or more slaves by invoking the Given slave address or addresses. All of the slaves may be contacted by using the Broadcast address. Two special Function Registers are used to define the slave's address, SADDR, and the address mask, SADEN. SADEN is used to define which bits in the SADDR are to be used and which bits are "don't care". The SADEN mask can be logically ANDed with the SADDR to create the "Given" address which the master will use for addressing each of the slaves. Use of the Given address allows multiple slaves to be recognized while excluding others. The following examples will help to show the versatility of this scheme:

Slave 0	SADDR =	1100 0000
	SADEN =	1111 1101
	Given =	1100 00X0

Slave 1	SADDR =	1100 0000
	SADEN =	1111 1110
	Given =	1100 000X

In the above example SADDR is the same and the SADEN data is used to differentiate between the two slaves. Slave 0 requires a 0 in bit 0 and it ignores bit 1. Slave 1 requires a 0 in bit 1 and bit 0 is ignored. A unique address for Slave 0 would be 1100 0010 since slave 1 requires a 0 in bit 1. A unique address for slave 1 would be 1100 0001 since a 1 in bit 0 will exclude slave 0. Both slaves can be selected at the same time by an address which has bit 0 = 0 (for slave 0) and bit 1 = 0 (for slave 1). Thus, both could be addressed with 1100 0000.

In a more complex system the following could be used to select slaves 1 and 2 while excluding slave 0:

Slave 0	SADDR =	1100 0000
	SADEN =	1111 1001
	Given =	1100 0XX0

Slave 1	SADDR =	1110 0000
	SADEN =	1111 1010
	Given =	1110 0X0X

Slave 2	SADDR =	1110 0000
	SADEN =	1111 1100
	Given =	1110 00XX

In the above example the differentiation among the 3 slaves is in the lower 3 address bits. Slave 0 requires that bit 0 = 0 and it can be uniquely addressed by 1110 0110. Slave 1 requires that bit 1 = 0 and it can be uniquely addressed by 1110 and 0101. Slave 2 requires that bit 2 = 0 and its unique address is 1110 0011. To select Slaves 0 and 1 and exclude Slave 2 use address 1110 0100, since it is necessary to make bit 2 = 1 to exclude slave 2.

The Broadcast Address for each slave is created by taking the logical OR of SADDR and SADEN. Zeros in this result are treated as don't-cares. In most cases, interpreting the don't-cares as ones, the broadcast address will be FF hexadecimal.

Upon reset SADDR (SFR address 0A9H) and SADEN (SFR address 0B9H) are loaded with 0s. This produces a given address of all "don't cares" as well as a Broadcast address of all "don't cares". This effectively disables the Automatic Addressing mode and allows the microcontroller to use standard 80C51 type UART drivers which do not make use of this feature.

**Reduced EMI Mode**

The AO bit (AUXR.O) in the AUXR register when set disables the ALE output.

**8XC51FC Reduced EMI Mode****AUXR (0X8E)**

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	AO

AO: Turns off ALE output.

**Interrupt Priority Structure**

The 8XC51FC has a 7-source four-level interrupt structure. There are 3 SFRs associated with the interrupts on the 8XC51FC. They are the IE and IP which are identical in function to those on the 80C51. In addition, there is the IPH (Interrupt Priority High) register that makes the four-level interrupt structure possible. The IPH is bit addressable and is located at SFR address B7H. The structure of the IPH register and a description of its bits is shown below:

**IPH (Interrupt Priority High) (B7H)**

7	6	5	4	3	2	1	0
—	PPCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H

IPH.0	PX0H	External interrupt 0 priority high
IPH.1	PT0H	Timer 0 interrupt priority high
IPH.2	PX1H	External interrupt 1 priority high
IPH.3	PT1H	Timer 1 interrupt priority high
IPH.4	PSH	Serial Port interrupt high
IPH.5	PT2H	Timer 2 interrupt priority high
IPH.6	PPCH	PCA interrupt priority high
IPH.7	—	Not implemented



## CMOS single-chip 8-bit microcontrollers 83C51FC/87C51FC

The function of the IPH SFR is simple and when combined with the IP SFR determines the priority of each interrupt. The priority of each interrupt is determined as shown in the following table:

PRIORITY BITS		INTERRUPT PRIORITY LEVEL
IPH.x	IP.x	
0	0	Level 0 (lowest priority)
0	1	Level 1
1	0	Level 2
1	1	Level 3 (highest priority)

The priority scheme for servicing the interrupts is the same as that for the 80C51, except there are four interrupt levels on the 8XC51FC rather than two as on the 80C51.

An interrupt will be serviced as long as an interrupt of equal or higher priority is not already being serviced. If an interrupt of equal or higher level priority is being

served, the new interrupt will wait until it is finished before being serviced. If a lower priority level interrupt is being serviced, it will be stopped and the new interrupt serviced.

When the new interrupt is finished, the lower priority level interrupt that was stopped will be completed.

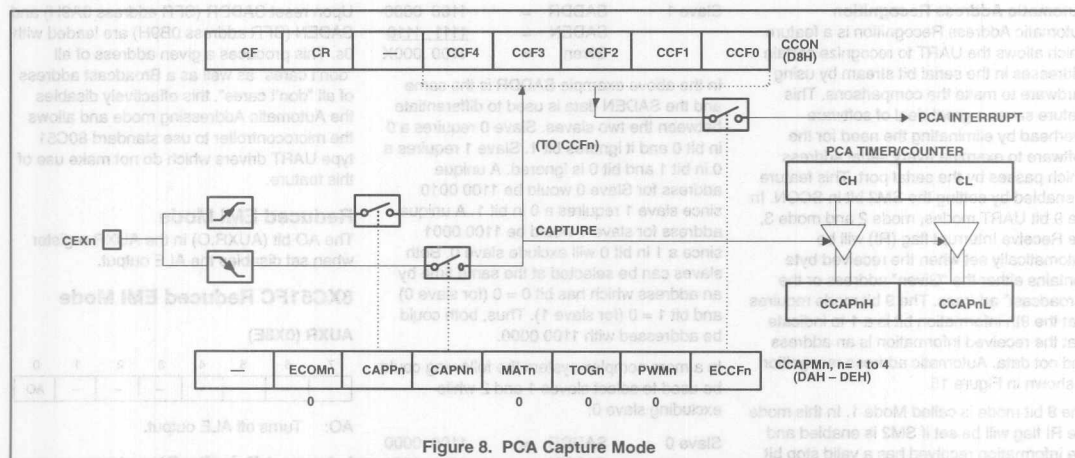


Figure 8. PCA Capture Mode

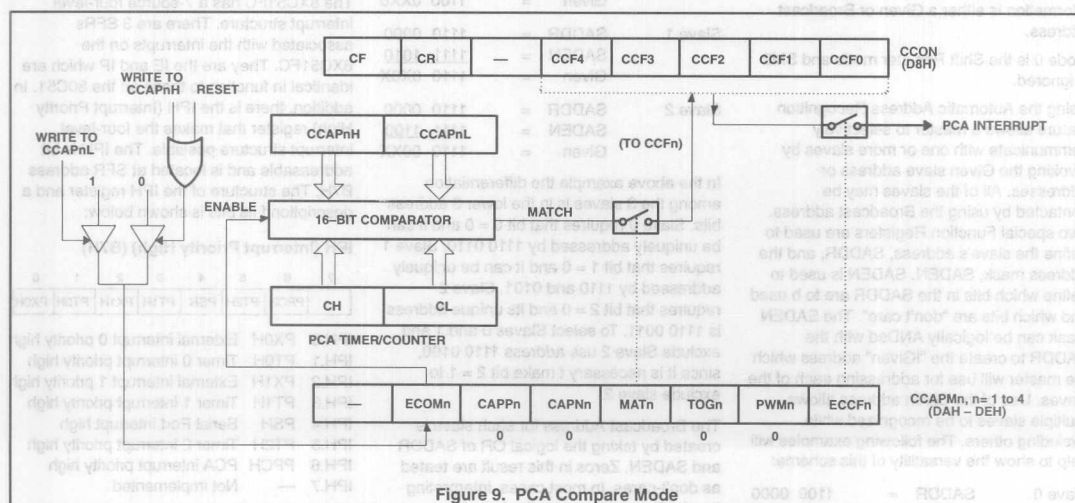
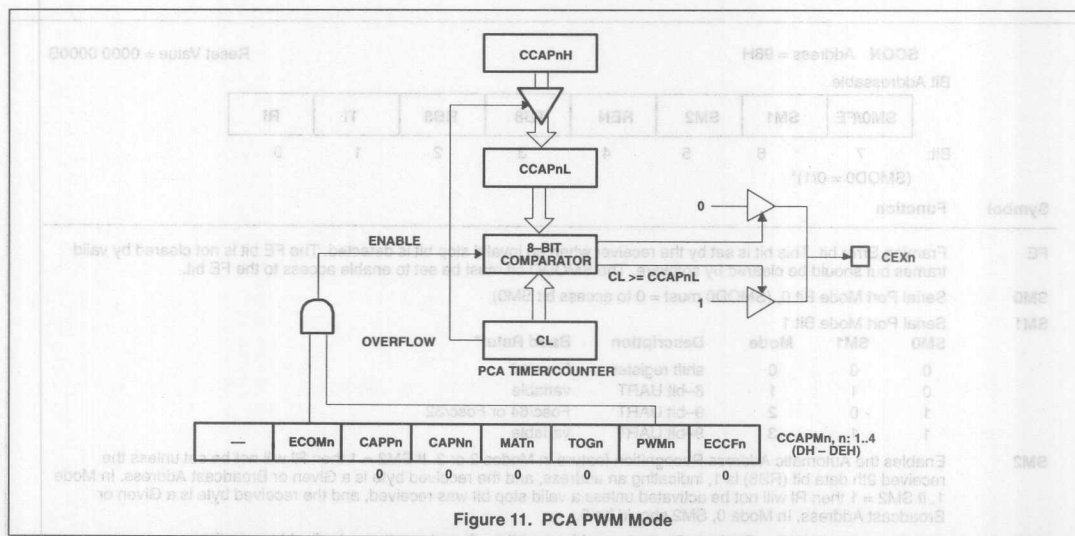
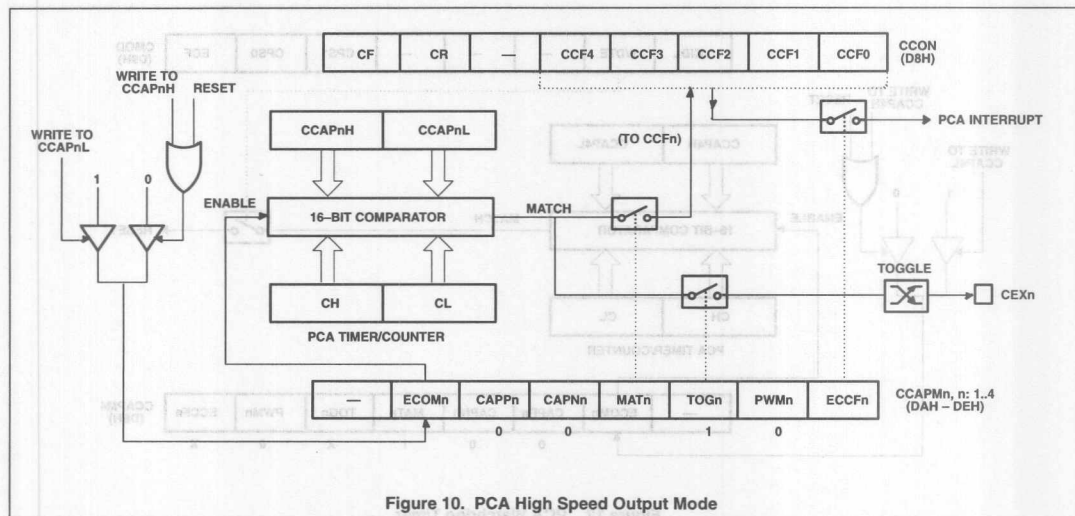
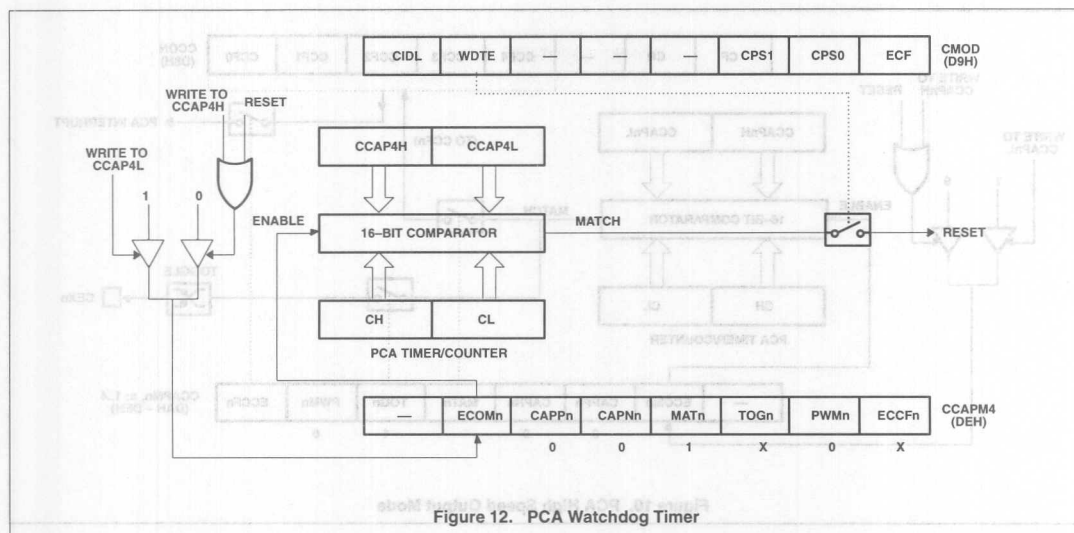


Figure 9. PCA Compare Mode

## CMOS single-chip 8-bit microcontrollers 83C51FC/87C51FC



## CMOS single-chip 8-bit microcontrollers 83C51FC/87C51FC



SCON Address = 98H

Reset Value = 0000 0000B

Bit Addressable

SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI
--------	-----	-----	-----	-----	-----	----	----

Bit: 7 6 5 4 3 2 1 0

(SMOD0 = 0/1)\*

Symbol

Function

<b>FE</b>	Framing Error bit. This bit is set by the receiver when an invalid stop bit is detected. The FE bit is not cleared by valid frames but should be cleared by software. The SMOD0 bit must be set to enable access to the FE bit.								
<b>SM0</b>	Serial Port Mode Bit 0, (SMOD0 must = 0 to access bit SM0)								
<b>SM1</b>	Serial Port Mode Bit 1								
	<b>SM0</b>	<b>SM1</b>	<b>Mode</b>	<b>Description</b>	<b>Baud Rate**</b>				
	0	0	0	shift register	Fosc/12				
	0	1	1	8-bit UART	variable				
	1	0	2	9-bit UART	Fosc/64 or Fosc/32				
	1	1	3	9-bit UART	variable				
<b>SM2</b>	Enables the Automatic Address Recognition feature in Modes 2 or 3. If SM2 = 1 then RI will not be set unless the received 9th data bit (RB8) is 1, indicating an address, and the received byte is a Given or Broadcast Address. In Mode 1, if SM2 = 1 then RI will not be activated unless a valid stop bit was received, and the received byte is a Given or Broadcast Address. In Mode 0, SM2 should be 0.								
<b>REN</b>	Enables serial reception. Set by software to enable reception. Clear by software to disable reception.								
<b>TB8</b>	The 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired.								
<b>RB8</b>	In modes 2 and 3, the 9th data bit that was received. In Mode 1, if SM2 = 0, RB8 is the stop bit that was received. In Mode 0, RB8 is not used.								
<b>TI</b>	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.								
<b>RI</b>	Receive interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM2). Must be cleared by software.								

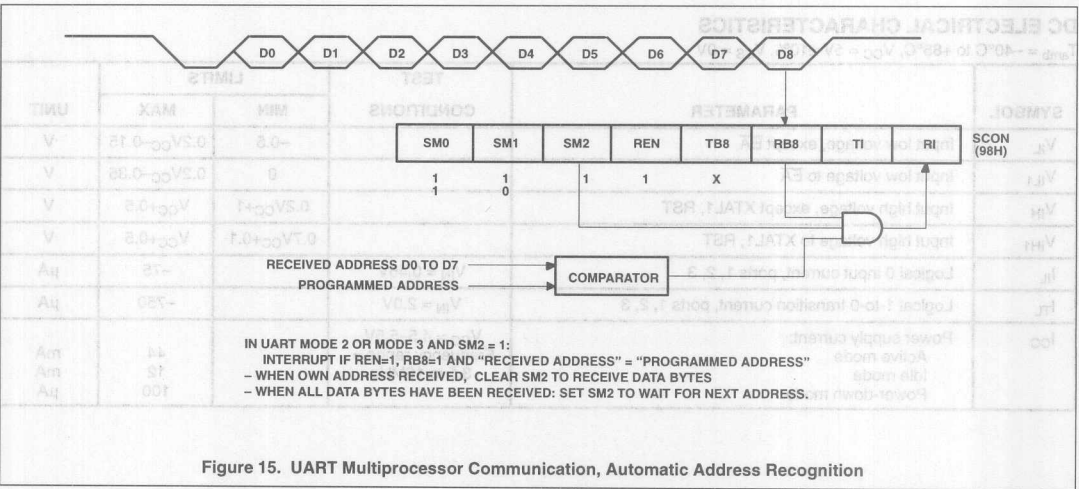
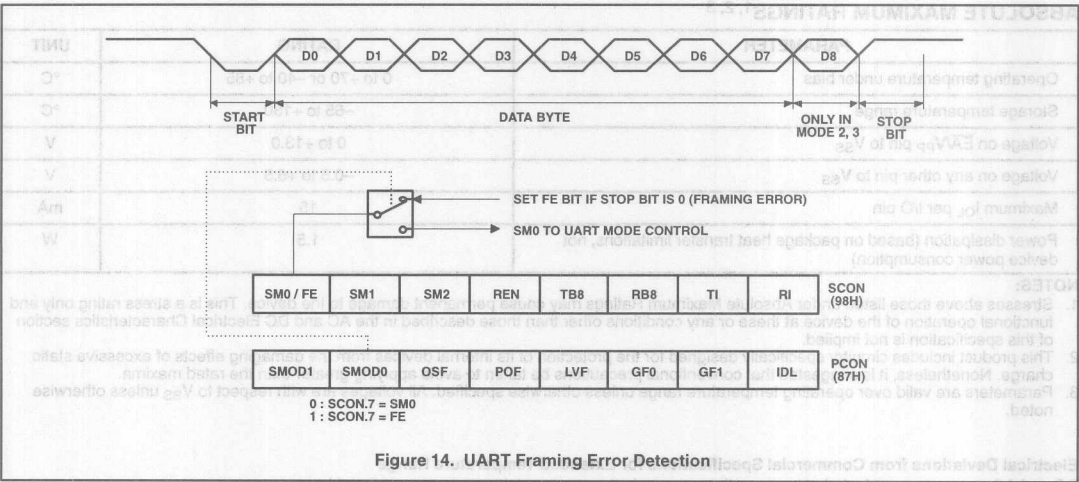
NOTE:

\*SMOD0 is located at PCON6.

\*\*Fosc = oscillator frequency

Figure 13. SCON: Serial Port Control Register

CMOS single-chip 8-bit microcontrollers 83C51FC/87C51FC



## CMOS single-chip 8-bit microcontrollers 83C51FC/87C51FC

ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>

PARAMETER	RATING	UNIT
Operating temperature under bias	0 to +70 or -40 to +85	°C
Storage temperature range	-65 to +150	°C
Voltage on $\overline{EA}/V_{PP}$ pin to $V_{SS}$	0 to +13.0	V
Voltage on any other pin to $V_{SS}$	-0.5 to +6.5	V
Maximum $I_{OL}$ per I/O pin	15	mA
Power dissipation (based on package heat transfer limitations, not device power consumption)	1.5	W

## NOTES:

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
- This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
- Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to  $V_{SS}$  unless otherwise noted.

## Electrical Deviations from Commercial Specifications for Extended Temperature Range

DC and AC parameters not included here are the same as in the commercial temperature range table.

## DC ELECTRICAL CHARACTERISTICS

$T_{amb} = -40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			MIN	MAX	
$V_{IL}$	Input low voltage, except $\overline{EA}$		-0.5	$0.2V_{CC}-0.15$	V
$V_{IL1}$	Input low voltage to $\overline{EA}$		0	$0.2V_{CC}-0.35$	V
$V_{IH}$	Input high voltage, except XTAL1, RST		$0.2V_{CC}+1$	$V_{CC}+0.5$	V
$V_{IH1}$	Input high voltage to XTAL1, RST		$0.7V_{CC}+0.1$	$V_{CC}+0.5$	V
$I_{IL}$	Logical 0 input current, ports 1, 2, 3	$V_{IN} = 0.45\text{V}$		-75	$\mu\text{A}$
$I_{TL}$	Logical 1-to-0 transition current, ports 1, 2, 3	$V_{IN} = 2.0\text{V}$		-750	$\mu\text{A}$
$I_{CC}$	Power supply current: Active mode Idle mode Power-down mode	$V_{CC} = 4.5-5.5\text{V}$ , Frequency range = 3.5 to 16MHz		44 12 100	mA mA $\mu\text{A}$



## CMOS single-chip 8-bit microcontrollers 83C51FC/87C51FC

## DC ELECTRICAL CHARACTERISTICS

 $T_{amb} = 0^{\circ}\text{C to } +70^{\circ}\text{C or } -40^{\circ}\text{C to } +85^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$ 

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP <sup>1</sup>	MAX	
$V_{IL}$	Input low voltage, except EA <sup>7</sup>		-0.5		$0.2V_{CC} - 0.1$	V
$V_{IL1}$	Input low voltage to EA <sup>7</sup>		0		$0.2V_{CC} - 0.3$	V
$V_{IH}$	Input high voltage, except XTAL1, RST <sup>7</sup>		$0.2V_{CC} + 0.9$		$V_{CC} + 0.5$	V
$V_{IH1}$	Input high voltage, XTAL1, RST <sup>7</sup>		$0.7V_{CC}$		$V_{CC} + 0.5$	V
$V_{OL}$	Output low voltage, ports 1, 2, 3 <sup>9</sup>	$I_{OL} = 100\mu\text{A}$ $I_{OL} = 1.6\text{mA}$ $I_{OL} = 3.5\text{mA}$			0.3 0.45 1.0	V V V
$V_{OL1}$	Output low voltage, port 0, ALE, PSEN <sup>9</sup>	$I_{OL} = 200\mu\text{A}$ $I_{OL} = 3.2\text{mA}$ $I_{OL} = 7.0\text{mA}$			0.3 0.45 1.0	V V V
$V_{OH}$	Output high voltage, ports 1, 2, 3, ALE, PSEN <sup>3</sup>	$I_{OH} = -60\mu\text{A}$ , $I_{OH} = -30\mu\text{A}$ $I_{OH} = -10\mu\text{A}$	$V_{CC} - 1.5$ $V_{CC} - 0.7$ $V_{CC} - 0.3$			V V V
$V_{OH1}$	Output high voltage (port 0 in external bus mode), ALE <sup>10</sup> , PSEN <sup>3</sup>	$I_{OH} = -7.0\text{mA}$ , $I_{OH} = -3.2\text{mA}$ $I_{OH} = -200\mu\text{A}$	$V_{CC} - 1.5$ $V_{CC} - 0.7$ $V_{CC} - 0.3$			V V V
$I_{IL}$	Logical 0 input current, ports 1, 2, 3 <sup>7</sup>	$V_{IN} = 0.45\text{V}$			-50	$\mu\text{A}$
$I_{TL}$	Logical 1-to-0 transition current, ports 1, 2, 3 <sup>7</sup>	See note 4			-650	$\mu\text{A}$
$I_{LI}$	Input leakage current, port 0	$0.45 V_{IN} < V_{CC} - 0.3$			$\pm 10$	$\mu\text{A}$
$I_{CC}$	Power supply current: <sup>7</sup> Active mode @ 16MHz <sup>5</sup> Idle mode @ 16MHz Power-down mode	See note 6		15 3 10	40 10 100	mA mA $\mu\text{A}$
$R_{RST}$	Internal reset pull-down resistor		50		225	k $\Omega$
$C_{IO}$	Pin capacitance <sup>11</sup> (except EA)				15	pF

## NOTES:

- Typical ratings are not guaranteed. The values listed are at room temperature, 5V.
- Capacitive loading on ports 0 and 2 may cause spurious noise to be superimposed on the  $V_{OL}$ s of ALE and ports 1 and 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100pF), the noise pulse on the ALE pin may exceed 0.8V. In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.  $I_{OL}$  can exceed these conditions provided that no single output sinks more than 5mA and no more than two outputs exceed the test conditions.
- Capacitive loading on ports 0 and 2 may cause the  $V_{OH}$  on ALE and PSEN to momentarily fall below the  $0.9V_{CC}$  specification when the address bits are stabilizing.
- Pins of ports 1, 2 and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when  $V_{IN}$  is approximately 2V.
- $I_{CCMAX}$  at other frequencies is given by: Active mode:  $I_{CCMAX} = 1.50 \times \text{FREQ} + 8$ ; Idle mode:  $I_{CCMAX} = 0.14 \times \text{FREQ} + 2.31$ , where FREQ is the external oscillator frequency in MHz.  $I_{CCMAX}$  is given in mA. See Figure 23.
- See Figures 24 through 27 for  $I_{CC}$  test conditions.
- These values apply only to  $T_{amb} = 0^{\circ}\text{C to } +70^{\circ}\text{C}$ . For  $T_{amb} = -40^{\circ}\text{C to } +85^{\circ}\text{C}$ , see table on previous page.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:  
Maximum  $I_{OL}$  per port pin: 15mA (\*NOTE: This is 85°C specification.)  
Maximum  $I_{OL}$  per 8-bit port: 26mA  
Maximum total  $I_{OL}$  for all outputs: 71mA  
If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
- ALE is tested to  $V_{OH1}$ , except when ALE is off then  $V_{OH}$  is the voltage specification.
- Pin capacitance is less than 25pF. Pin capacitance of ceramic package is less than 15pF (except EA it is 25pF).



## CMOS single-chip 8-bit microcontrollers 83C51FC/87C51FC

## AC ELECTRICAL CHARACTERISTICS

 $T_{amb} = 0^{\circ}\text{C to } +70^{\circ}\text{C or } -40^{\circ}\text{C to } +85^{\circ}\text{C}, V_{CC} = 5\text{V} \pm 10\%, V_{SS} = 0\text{V}^{1, 2, 3}$ 

SYMBOL	FIGURE	PARAMETER	16MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
$1/t_{CLCL}$	16	Oscillator frequency			3.5 3.5	16 24	MHz MHz
$t_{LHLL}$	16	ALE pulse width	85		$2t_{CLCL}-40$		ns
$t_{AVLL}$	16	Address valid to ALE low	22		$t_{CLCL}-40$		ns
$t_{LLAX}$	16	Address hold after ALE low	32		$t_{CLCL}-30$		ns
$t_{LLIV}$	16	ALE low to valid instruction in		150	$4t_{CLCL}-100$		ns
$t_{LLPL}$	16	ALE low to PSEN low	32		$t_{CLCL}-30$		ns
$t_{PLPH}$	16	PSEN pulse width	142		$3t_{CLCL}-45$		ns
$t_{PLIV}$	16	PSEN low to valid instruction in		82		$3t_{CLCL}-105$	ns
$t_{PXIX}$	16	Input instruction hold after PSEN	0		0		ns
$t_{PXIZ}$	16	Input instruction float after PSEN		37		$t_{CLCL}-25$	ns
$t_{AVIV}$	16	Address to valid instruction in		207		$5t_{CLCL}-105$	ns
$t_{PLAZ}$	16	PSEN low to address float		10		10	ns
<b>Data Memory</b>							
$t_{RLRH}$	17, 18	RD pulse width	275		$6t_{CLCL}-100$		ns
$t_{WLWH}$	17, 18	WR pulse width	275		$6t_{CLCL}-100$		ns
$t_{RLDV}$	17, 18	RD low to valid data in		147		$5t_{CLCL}-165$	ns
$t_{RHDZ}$	17, 18	Data hold after RD	0		0		ns
$t_{RHDZ}$	17, 18	Data float after RD		65		$2t_{CLCL}-60$	ns
$t_{LLDV}$	17, 18	ALE low to valid data in		350		$8t_{CLCL}-150$	ns
$t_{AVDV}$	17, 18	Address to valid data in		397		$9t_{CLCL}-165$	ns
$t_{LLWL}$	17, 18	ALE low to RD or WR low	137	237	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
$t_{AVWL}$	17, 18	Address valid to WR low or RD low	175		$4t_{CLCL}-75$		ns
$t_{QVWX}$	17, 18	Data valid to WR transition	42		$t_{CLCL}-20$		ns
$t_{WHQX}$	17, 18	Data hold after WR	42		$t_{CLCL}-20$		ns
$t_{QVWH}$	18	Data valid to WR high	287		$7t_{CLCL}-150$		ns
$t_{RLAZ}$	17, 18	RD low to address float		0		0	ns
$t_{WLHL}$	17, 18	RD or WR high to ALE high	40	87	$t_{CLCL}-20$	$t_{CLCL}+25$	ns
<b>External Clock</b>							
$t_{CHCX}$	20	High time	12		20		ns
$t_{CLCX}$	20	Low time	12		20		ns
$t_{CLCH}$	20	Rise time		20		20	ns
$t_{CHCL}$	20	Fall time		20		20	ns
<b>Shift Register</b>							
$t_{XLXL}$	19	Serial port clock cycle time	1		$12t_{CLCL}$		$\mu\text{s}$
$t_{QVXH}$	19	Output data setup to clock rising edge	492		$10t_{CLCL}-133$		ns
$t_{XHGX}$	19	Output data hold after clock rising edge	8		$2t_{CLCL}-117$		ns
$t_{XHDX}$	19	Input data hold after clock rising edge	0		0		ns
$t_{XHDV}$	19	Clock rising edge to input data valid		492		$10t_{CLCL}-133$	ns

## NOTES:

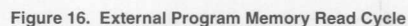
- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- Interfacing the 83C51FC to devices with float times up to 45ns is permitted. This limited bus contention will not cause damage to Port 0 drivers.

## 83C51FC/87C51FC

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

P – PSEN  
Q – Output data  
R – RD signal  
t – Time  
V – Valid  
W – WR signal  
X – No longer a valid logic level  
Z – Float

$t_{LLPL}$  = Time for ALE low to PSEN low.



## CMOS single-chip 8-bit microcontrollers 83C51FC/87C51FC

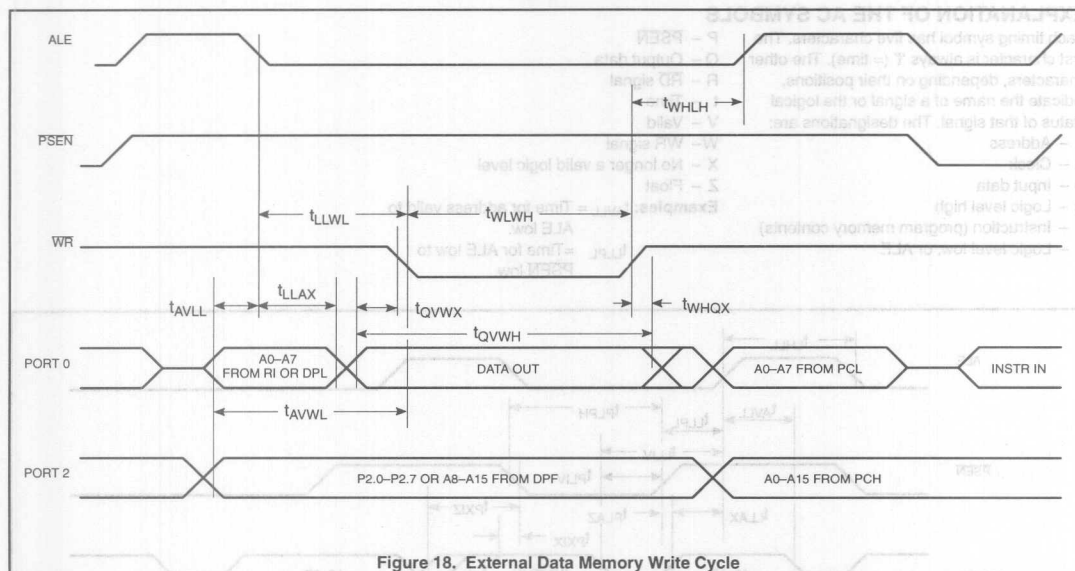


Figure 18. External Data Memory Write Cycle

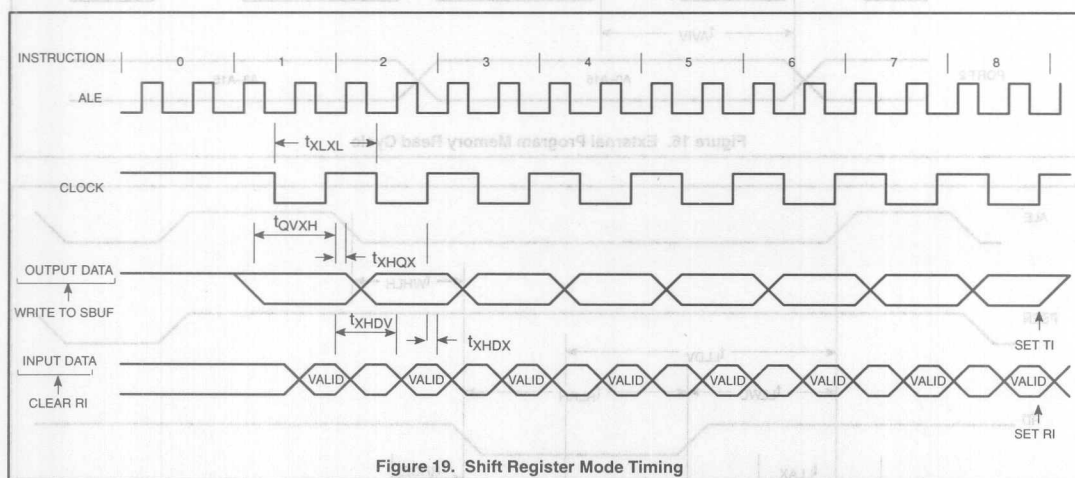


Figure 19. Shift Register Mode Timing

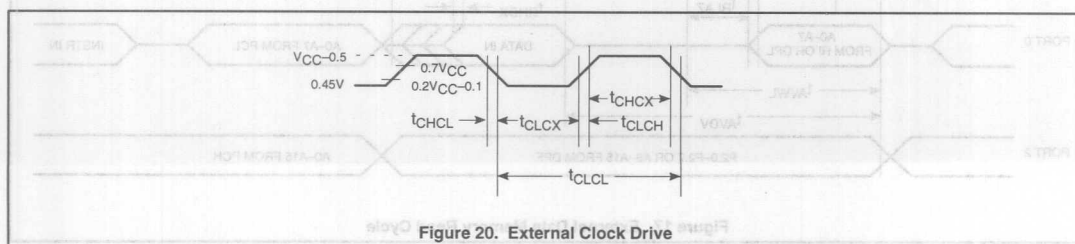
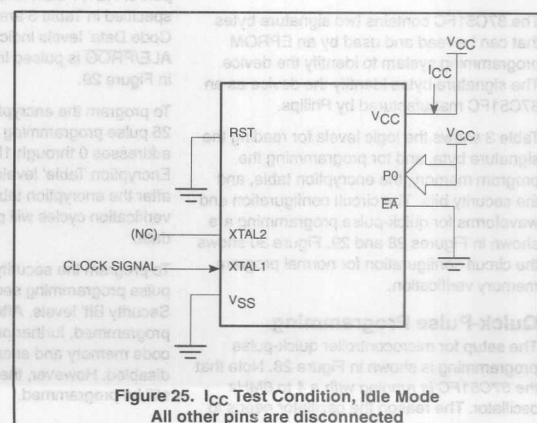
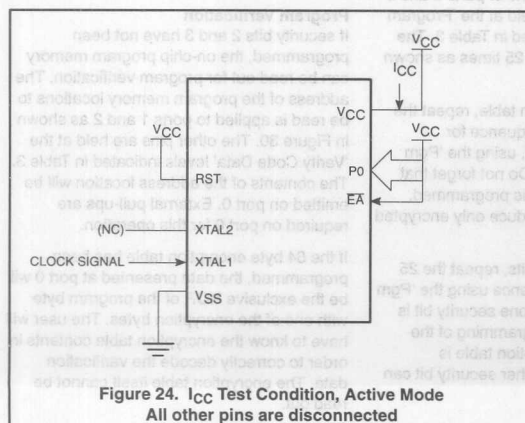
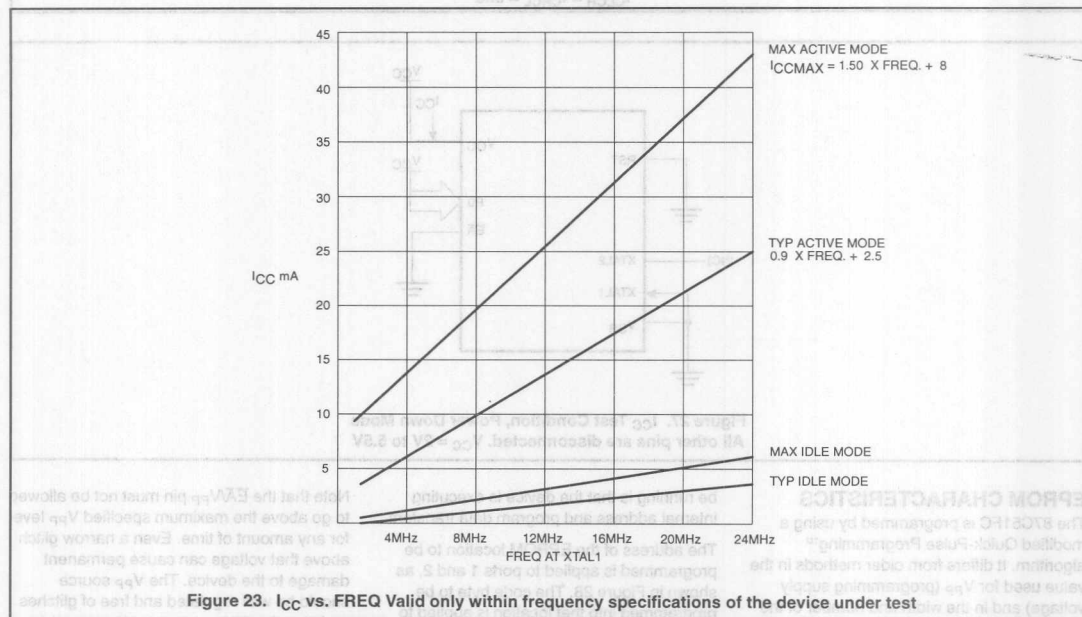
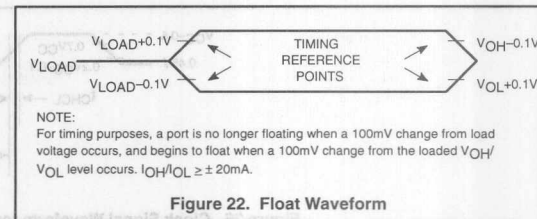
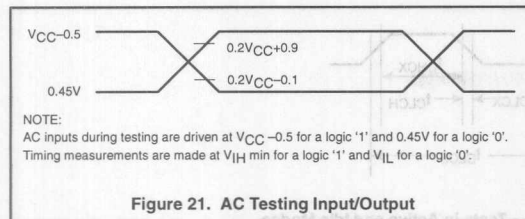


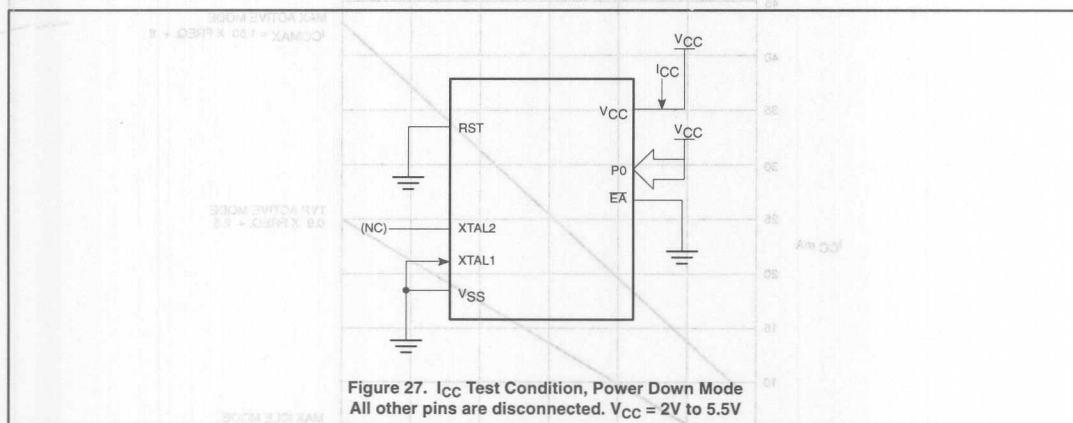
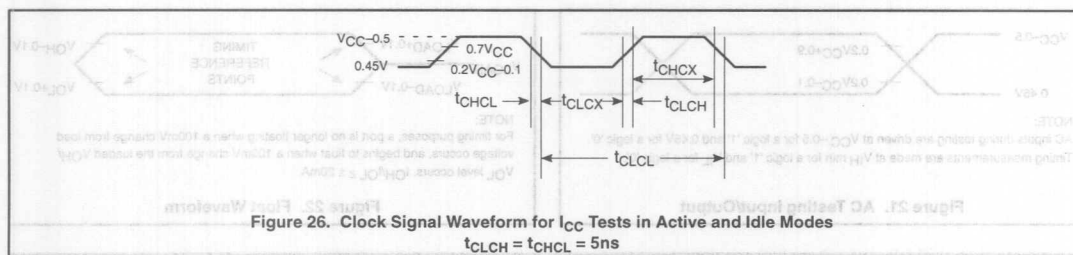
Figure 20. External Clock Drive

## CMOS single-chip 8-bit microcontrollers

## 83C51FC/87C51FC



## CMOS single-chip 8-bit microcontrollers 83C51FC/87C51FC

**EPROM CHARACTERISTICS**

The 87C51FC is programmed by using a modified Quick-Pulse Programming™ algorithm. It differs from older methods in the value used for  $V_{PP}$  (programming supply voltage) and in the width and number of the ALE/PROG pulses.

The 87C51FC contains two signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes identify the device as an 87C51FC manufactured by Philips.

Table 3 shows the logic levels for reading the signature byte, and for programming the program memory, the encryption table, and the security bits. The circuit configuration and waveforms for quick-pulse programming are shown in Figures 28 and 29. Figure 30 shows the circuit configuration for normal program memory verification.

**Quick-Pulse Programming**

The setup for microcontroller quick-pulse programming is shown in Figure 28. Note that the 87C51FC is running with a 4 to 6MHz oscillator. The reason the oscillator needs to

be running is that the device is executing internal address and program data transfers.

The address of the EPROM location to be programmed is applied to ports 1 and 2, as shown in Figure 28. The code byte to be programmed into that location is applied to port 0. RST, PSEN and pins of ports 2 and 3 specified in Table 3 are held at the 'Program Code Data' levels indicated in Table 3. The ALE/PROG is pulsed low 25 times as shown in Figure 29.

To program the encryption table, repeat the 25 pulse programming sequence for addresses 0 through 1FH, using the 'Pgm Encryption Table' levels. Do not forget that after the encryption table is programmed, verification cycles will produce only encrypted data.

To program the security bits, repeat the 25 pulse programming sequence using the 'Pgm Security Bit' levels. After one security bit is programmed, further programming of the code memory and encryption table is disabled. However, the other security bit can still be programmed.

Note that the  $\overline{EA}/V_{PP}$  pin must not be allowed to go above the maximum specified  $V_{PP}$  level for any amount of time. Even a narrow glitch above that voltage can cause permanent damage to the device. The  $V_{PP}$  source should be well regulated and free of glitches and overshoot.

**Program Verification**

If security bits 2 and 3 have not been programmed, the on-chip program memory can be read out for program verification. The address of the program memory locations to be read is applied to ports 1 and 2 as shown in Figure 30. The other pins are held at the 'Verify Code Data' levels indicated in Table 3. The contents of the address location will be emitted on port 0. External pull-ups are required on port 0 for this operation.

If the 64 byte encryption table has been programmed, the data presented at port 0 will be the exclusive NOR of the program byte with one of the encryption bytes. The user will have to know the encryption table contents in order to correctly decode the verification data. The encryption table itself cannot be read out.

™Trademark phrase of Intel Corporation.



## CMOS single-chip 8-bit microcontrollers 83C51FC/87C51FC

## Reading the Signature Bytes

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. The values are:

(030H) = 15H indicates manufactured by Philips

(031H) = B3H indicates 87C51FC

## Program/Verify Algorithms

Any algorithm in agreement with the conditions listed in Table 3, and which satisfies the timing specifications, is suitable.

## Erasure Characteristics

Erasure of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent erasure. **For this and secondary effects, it is recommended that an opaque label be placed over the window.** For elevated temperature or environments where solvents are being used, apply Kapton tape Fluorglas part number 2345-5, or equivalent.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 angstroms) to an integrated dose of at least 15W-s/cm<sup>2</sup>. Exposing the EPROM to an ultraviolet lamp of 12,000μW/cm<sup>2</sup> rating for

20 to 39 minutes, at a distance of about 1 inch, should be sufficient.

Erasure leaves the array in an all 1s state.

## Security Bits

With none of the security bits programmed the code in the program memory can be verified. If the encryption table is programmed, the code will be encrypted when verified. When only security bit 1 is programmed, MOV<sub>C</sub> instructions executed from external program memory are disabled from fetching code bytes from the internal memory, EA is latched on Reset and all further programming of the EPROM is disabled. When security bits 1 and 2 are programmed, in addition to the above, verify mode is disabled. When all three security bits are programmed, all of the conditions above apply and all external program memory execution is disabled.

Table 3. EPROM Programming Modes

MODE	RST	PSEN	ALE/PROG	EA/V <sub>PP</sub>	P2.7	P2.6	P3.7	P3.6	P3.3
Read signature	1	0	1	1	0	0	0	0	0
Program code data	1	0	0*	V <sub>PP</sub>	1	0	1	1	1
Verify code data	1	0	1	1	0	0	1	1	0
Pgm encryption table	1	0	0*	V <sub>PP</sub>	1	0	1	0	1
Pgm security bit 1	1	0	0*	V <sub>PP</sub>	1	1	1	1	1
Pgm security bit 2	1	0	0*	V <sub>PP</sub>	1	1	0	0	1
Pgm security bit 3	1	0	0*	V <sub>PP</sub>	0	1	0	1	1

## NOTES:

1. '0' = Valid low for that pin, '1' = valid high for that pin.

2. V<sub>PP</sub> = 12.75V ±0.25V.

3. V<sub>CC</sub> = 5V±10% during programming and verification.

\* ALE/PROG receives 25 programming pulses while V<sub>PP</sub> is held at 12.75V. Each programming pulse is low for 100μs (±10μs) and high for a minimum of 10μs.

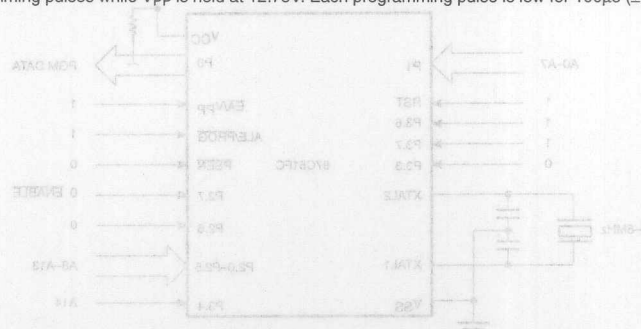
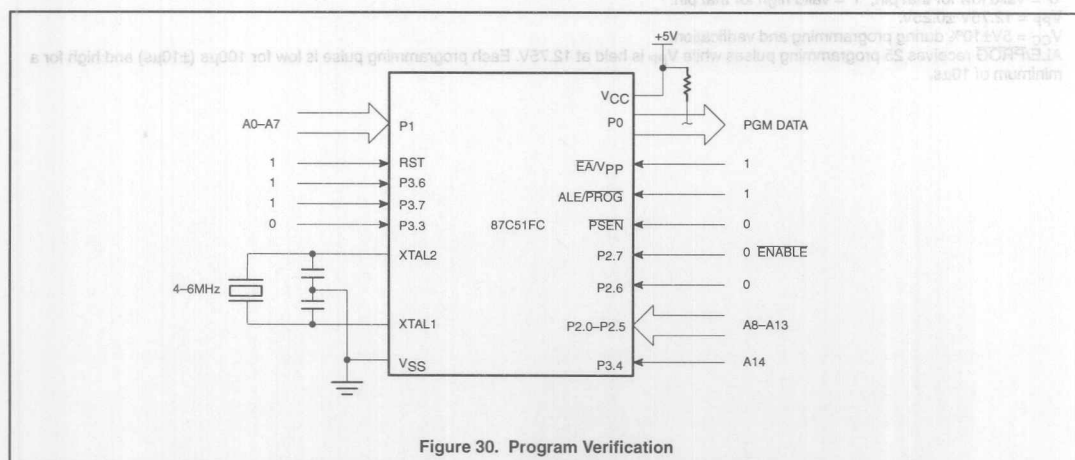
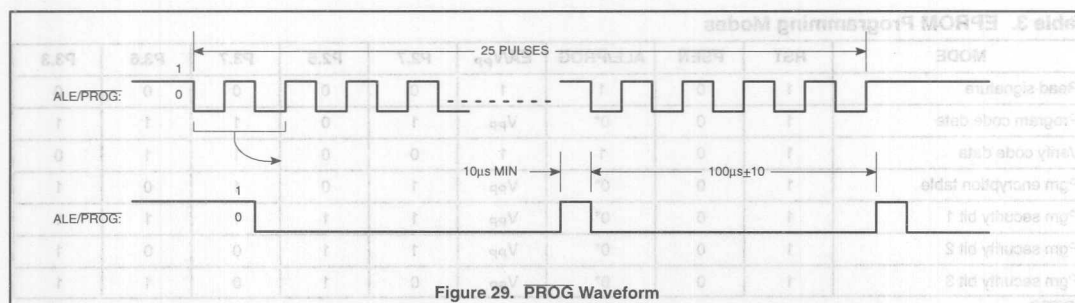
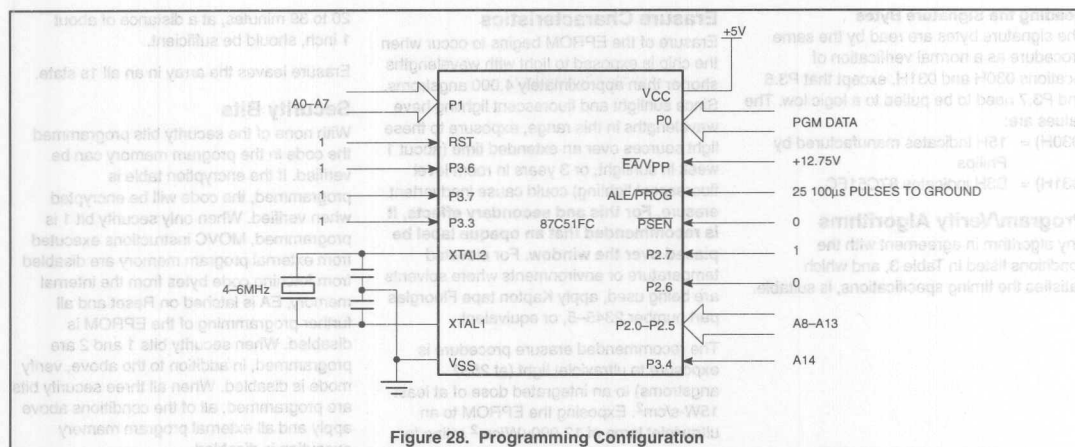


Figure 30. Program Verification



## CMOS single-chip 8-bit microcontrollers 83C51FC/87C51FC

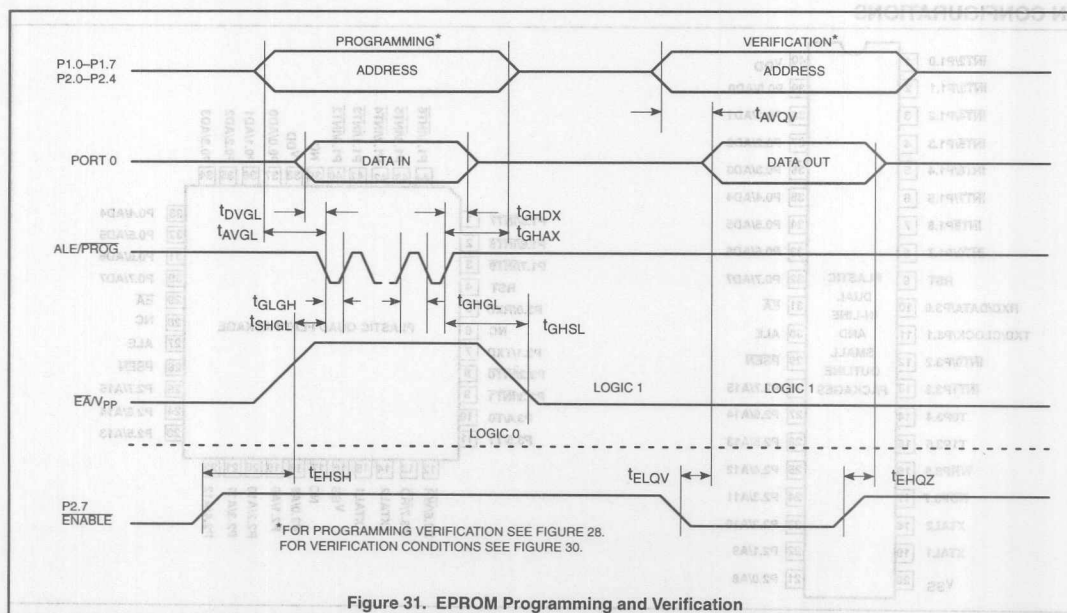


## CMOS single-chip 8-bit microcontrollers 83C51FC/87C51FC

## EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

$T_{amb} = 21^{\circ}\text{C}$  to  $+27^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$  (See Figure 31)

SYMBOL	PARAMETER	MIN	MAX	UNIT
$V_{PP}$	Programming supply voltage	12.5	13.0	V
$I_{PP}$	Programming supply current		50	mA
$1/t_{CLCL}$	Oscillator frequency	4	6	MHz
$t_{AVGL}$	Address setup to PROG low	$48t_{CLCL}$		
$t_{GHAX}$	Address hold after PROG	$48t_{CLCL}$		
$t_{DVGL}$	Data setup to PROG low	$48t_{CLCL}$		
$t_{GHDX}$	Data hold after PROG	$48t_{CLCL}$		
$t_{EHS}$	P2.7 (ENABLE) high to $V_{PP}$	$48t_{CLCL}$		
$t_{SHGL}$	$V_{PP}$ setup to PROG low	10		$\mu\text{s}$
$t_{GHSL}$	$V_{PP}$ hold after PROG	10		$\mu\text{s}$
$t_{GLGH}$	PROG width	90	110	$\mu\text{s}$
$t_{AVQV}$	Address to data valid		$48t_{CLCL}$	
$t_{ELOZ}$	ENABLE low to data valid		$48t_{CLCL}$	
$t_{EHQZ}$	Data float after ENABLE	0	$48t_{CLCL}$	
$t_{GHGL}$	PROG high to PROG low	10		$\mu\text{s}$



## Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

### FEATURES

- Full static 80C51 CPU
- 8-bit CPU, ROM, RAM, I/O in a single 40-lead DIL / mini-pack
- 4K x 8 ROM, expandable externally to 64K bytes
- 128 bytes RAM, expandable externally to 64K bytes
- Four 8-bit ports, 32 I/O lines
- Two 16-bit timer / event counters
- External memory expandable up to 128K, external ROM up to 64K and / or RAM up to 64K
- On-chip oscillator suitable for RC, LC, quartz crystal or ceramic resonator
- Thirteen source, thirteen vector interrupt structure with two priority levels
- Full duplex serial port (UART)

- Enhanced architecture with:
  - non-page oriented instructions
  - direct addressing
  - four eight byte RAM register banks
  - stack depth up to 128 bytes
  - multiply, divide, subtract and compare instructions
- Power-Down and IDLE instructions
- Wake-up via external interrupts at Port 1
- Single supply voltage of 1.8V to 6.0V (5.0V  $\pm$  10% for P80C51)
- Frequency range of 0 to 16MHz (3.5MHz to 16MHz for P80C51)
- Very low current consumption
- Operating temperature range: -40 to +85°C

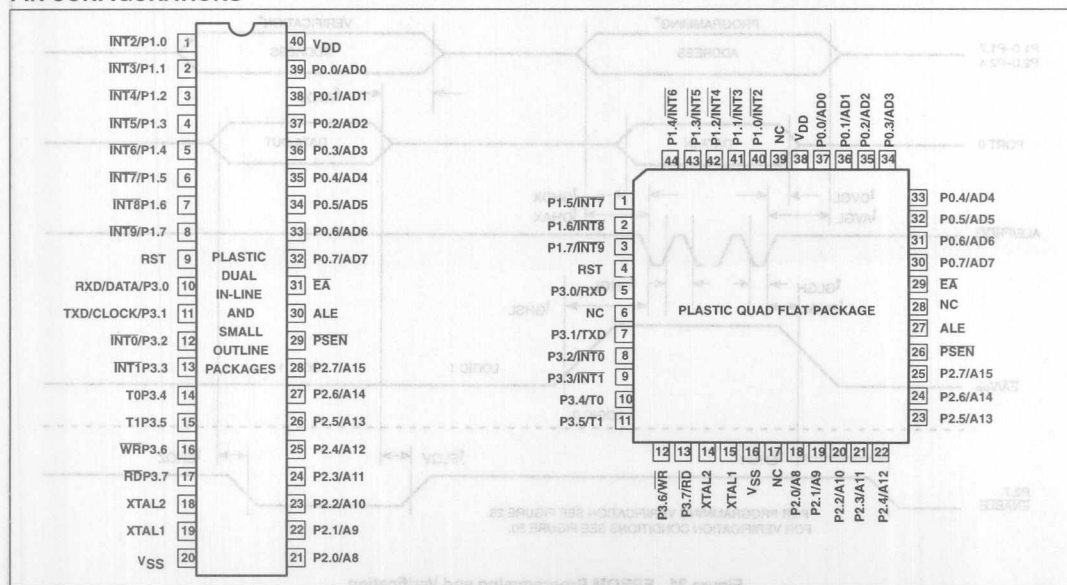
### DESCRIPTION

The 80CL51 is manufactured in an advanced CMOS technology. The instruction set of the 80CL51 is based on that of the 8051. The 80CL51 is a general purpose microcontroller especially suited for battery-powered applications. The device has low power consumption and a wide range of supply voltage. For emulation purposes, the 85CL000 (Piggy-back version) with 256 bytes of RAM is recommended. The 80CL51 has two software selectable modes of reduced activity for further power reduction: Idle and Power-down. The 80CL51 also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of over 100 instructions: 49 one-byte, 46 two-byte, and 16 three-byte.

The P80CL31 is the ROMless version of the P80CL51. P80C51 is a 5V version of the low voltage P80CL51.

The P80CL31 is the ROMless version of the P80CL51. P80C51 is a 5V version of the low voltage P80CL51.

### PIN CONFIGURATIONS



Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

ORDERING INFORMATION

PHILIPS PART ORDER NUMBER PART MARKING		PHILIPS NORTH AMERICA 1 PART ORDER NUMBER		TEMPERATURE RANGE °C AND PACKAGE	DRAWING NUMBER
ROMless	ROM	ROMless	ROM		
P80CL31HFP	P80CL51HFP	P80CL31HFP N	P80CL51HFP N	-40 to +85; 40-lead Plastic Dual In-line Package (1.8V to 6V)	SOT129
P80CL31HFT	P80CL51HFT	P80CL31HFT D	P80CL51HFT D	-40 to +85; 40-lead Plastic Small Outline Package (1.8V to 6V)	SOT158A
	P80CL51HFH	P80CL41HFH B	P80CL51HFH B	-40 to +85; 44-lead Plastic Quad Flat Package (1.8V to 6V)	SOT307
	P80C51HFP		P80C51HFP N	-40 to +85; 40-lead Plastic Dual In-line Package (5.0V ±10%)	SOT129
	P80C51HFT		P80C51HFT D	-40 to +85; 40-lead Plastic Small Outline Package (5.0V ±10%)	SOT158A
	P80C51HFH		P80C51HFH B	-40 to +85; 44-lead Plastic Quad Flat Package (5.0V ±10%)	SOT307

NOTE:

1. Parts ordered by the Philips North America part number will be marked with the Philips part marking.

# Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

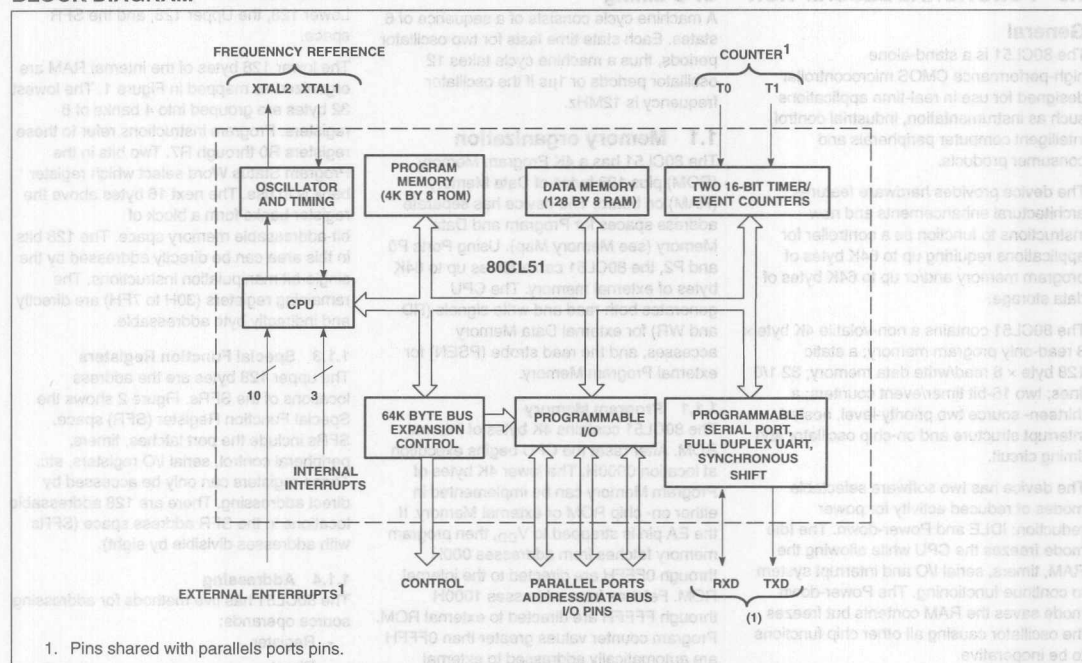
## PIN DESCRIPTIONS

PIN		DESIGNATION	FUNCTION
QFP	DIP		
40	1	P1.0/INT2	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pullups. Port 1 pins that have 1s written to them are pulled HIGH by the internal pullups, and in that state can be used as inputs. The Port 1 output buffer can sink/source 4 LS TTL loads. As inputs, Port 1 pins that are externally pulled LOW will source current ( $I_{IL}$ in the characteristics) due to the internal pullups. Port 1 also serves the alternative functions INT2 to INT9.
41	2	P1.1/INT3	
42	3	P1.2/INT4	
43	4	P1.3/INT5	
44	5	P1.4/INT6	
1	6	P1.5/INT7	
2	7	P1.6/INT8	
3	8	P1.7/INT9	
4	9	RST	<b>Reset:</b> A high level on this pin for two machine cycles while the oscillator is running resets the device.
5-13	10-17		<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 3 output buffers can sink/source 4 LS TTL inputs. Port 3 pins that have 1s written to them are pulled HIGH by the internal pull ups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally pulled LOW will source current ( $I_{IL}$ in the characteristics) due to the internal pull ups.
5	10	P3.0/RXD/data	<b>RXD/data:</b> Serial port receiver data input (asynchronous) or data input/output (synchronous)
7	11	P3.1/TXD/clock	<b>TXD/clock:</b> Serial port transmitter data output (asynchronous) or clock output (synchronous)
8	12	P3.2/INT0	<b>INT0:</b> External interrupt 0.
9	13	P3.3/INT1	<b>INT1:</b> External interrupt 1.
10	14	P3.4/T0	<b>T0:</b> Timer 0 external input.
11	15	P3.5/T1	<b>T1:</b> Timer 1 external input.
12	16	P3.6/WR	<b>WR:</b> External data memory write strobe.
13	17	P3.7/RD	<b>RD:</b> External data memory read strobe.
14	18	XTAL2	<b>Crystal output:</b> Output of the inverting amplifier of the oscillator. Left open when external clock is used. <b>Crystal input:</b> Input to the inverting amplifier of the oscillator; also the input for an externally generated clock source.
15	19	XTAL1	<b>Crystal input:</b> Input to the inverting amplifier of the oscillator; also the input for an externally generated clock source.
16	20	Vss	<b>Ground:</b> Circuit ground potential.
18-25	21-28	P2.0-P2.7	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pullups. Port 2 pins that have 1s written to them are pulled HIGH by the internal pullups, and in that state can be used as inputs. The Port 2 output buffer can sink/source 4 LS TTL loads. Port 2 emits the high-order address byte during accesses to external memory that use 1 6-bit addresses (MOVX @DPTR). In this application it uses the strong internal pullups when emitting 1s. During accesses to external memory that use 8-bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 Special Function Register.
26	29	PSEN	<b>Program store enable output:</b> Read strobe to external program memory. When executing code out of external program memory, PSEN is activated twice each machine cycle. However, during each access to external data memory two PSEN activations are skipped.
27	30	ALE	<b>Address Latch Enable:</b> Output pulse for latching the low byte of the address during access to external memory. ALE is emitted at a constant rate of 1/6 of the oscillator frequency, and may be used for external timing or clocking purposes.
29	31	EA	<b>External Access:</b> When EA is held High the CPU executes out of internal program memory (unless the program counter exceeds 0FFFH). Holding EA LOW forces the CPU to execute out of external memory regardless of the value of the program counter.
30-37	32-39	P0.0-P0.7	<b>Port 0:</b> Port 0 is an 8-bit open drain bidirectional I/O port. As an open drain output port it can sink 8 LS TTL loads. Port 0 pins that have 1s written to them float, and in that state will function as high impedance inputs. Port 0 is also the multiplexed low order address and data bus during access to external memory. In this application it uses strong internal pull-ups when emitting logic 1s.
38	40	VDD	<b>Power supply.</b>

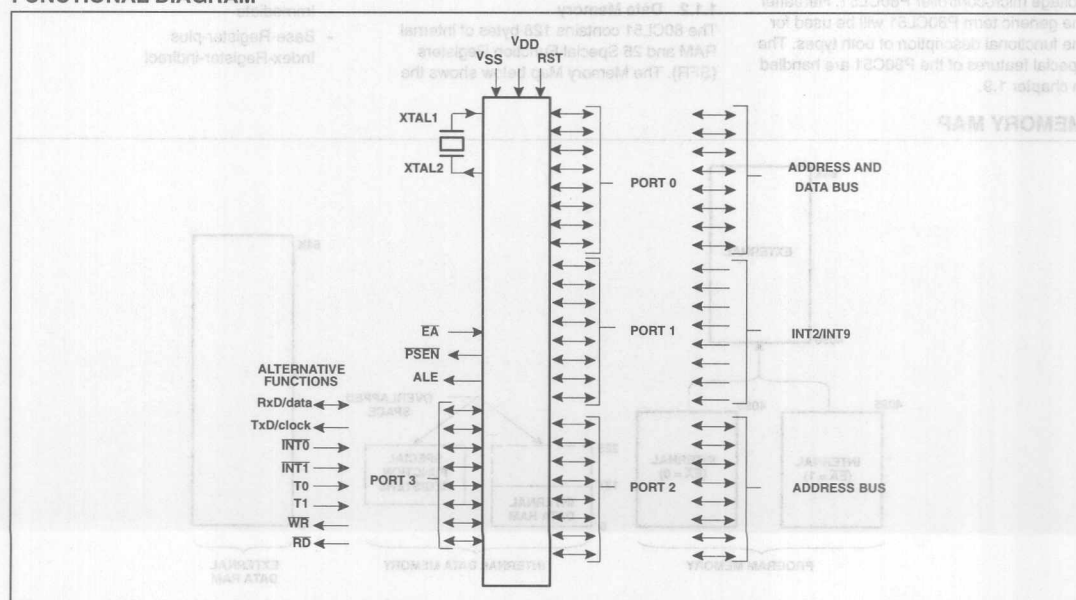


## Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

### BLOCK DIAGRAM



### FUNCTIONAL DIAGRAM





# Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

## 1.0 FUNCTIONAL DESCRIPTION

### General

The 80CL51 is a stand-alone high-performance CMOS microcontroller designed for use in real-time applications such as instrumentation, industrial control, intelligent computer peripherals and consumer products.

The device provides hardware features, architectural enhancements and new instructions to function as a controller for applications requiring up to 64K bytes of program memory and/or up to 64K bytes of data storage.

The 80CL51 contains a non-volatile 4K byte  $\times$  8 read-only program memory; a static 128 byte  $\times$  8 read/write data memory; 32 I/O lines; two 16-bit timer/event counters; a thirteen- source two priority-level, nested interrupt structure and on-chip oscillator and timing circuit.

The device has two software selectable modes of reduced activity for power reduction: IDLE and Power-down. The Idle mode freezes the CPU while allowing the RAM, timers, serial I/O and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator causing all other chip functions to be inoperative.

The P80C51 is a 5V version of the low voltage microcontroller P80CL51. Hereafter the generic term P80CL51 will be used for the functional description of both types. The special features of the P80C51 are handled in chapter 1.9.

### CPU timing

A machine cycle consists of a sequence of 6 states. Each state time lasts for two oscillator periods, thus a machine cycle takes 12 oscillator periods or 1  $\mu$ s if the oscillator frequency is 12MHz.

### 1.1 Memory organization

The 80CL51 has a 4K Program Memory (ROM) plus 128 bytes of Data Memory (RAM) on board. The device has separate address spaces for Program and Data Memory (see Memory Map). Using Ports P0 and P2, the 80CL51 can address up to 64K bytes of external memory. The CPU generates both read and write signals (RD and WR) for external Data Memory accesses, and the read strobe (PSEN) for external Program Memory.

#### 1.1.1 Program Memory

The 80CL51 contains 4K bytes of internal ROM. After reset the CPU begins execution at location 0000H. The lower 4K bytes of Program Memory can be implemented in either on-chip ROM or external Memory. If the EA pin is strapped to  $V_{DD}$ , then program memory fetches from addresses 000H through 0FFFH are directed to the internal ROM. Fetches from addresses 1000H through FFFFH are directed to external ROM. Program counter values greater than 0FFFH are automatically addressed to external memory regardless of the state of the EA pin.

#### 1.1.2 Data Memory

The 80CL51 contains 128 bytes of internal RAM and 25 Special Function Registers (SFR). The Memory Map below shows the

internal Data Memory space divided into the Lower 128, the Upper 128, and the SFR space.

The lower 128 bytes of the internal RAM are organized as mapped in Figure 1. The lowest 32 bytes are grouped into 4 banks of 8 registers. Program instructions refer to these registers R0 through R7. Two bits in the Program Status Word select which register bank is in use. The next 16 bytes above the register banks form a block of bit-addressable memory space. The 128 bits in this area can be directly addressed by the single-bit manipulation instructions. The remaining registers (30H to 7FH) are directly and indirectly byte addressable.

#### 1.1.3 Special Function Registers

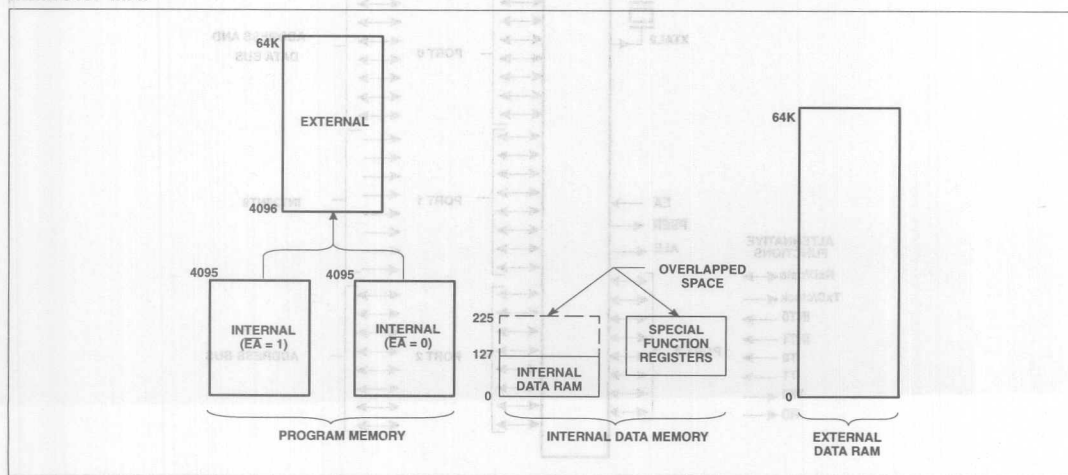
The upper 128 bytes are the address locations of the SFRs. Figure 2 shows the Special Function Register (SFR) space. SFRs include the port latches, timers, peripheral control, serial I/O registers, etc. These registers can only be accessed by direct addressing. There are 128 addressable locations in the SFR address space (SFRs with addresses divisible by eight).

#### 1.1.4 Addressing

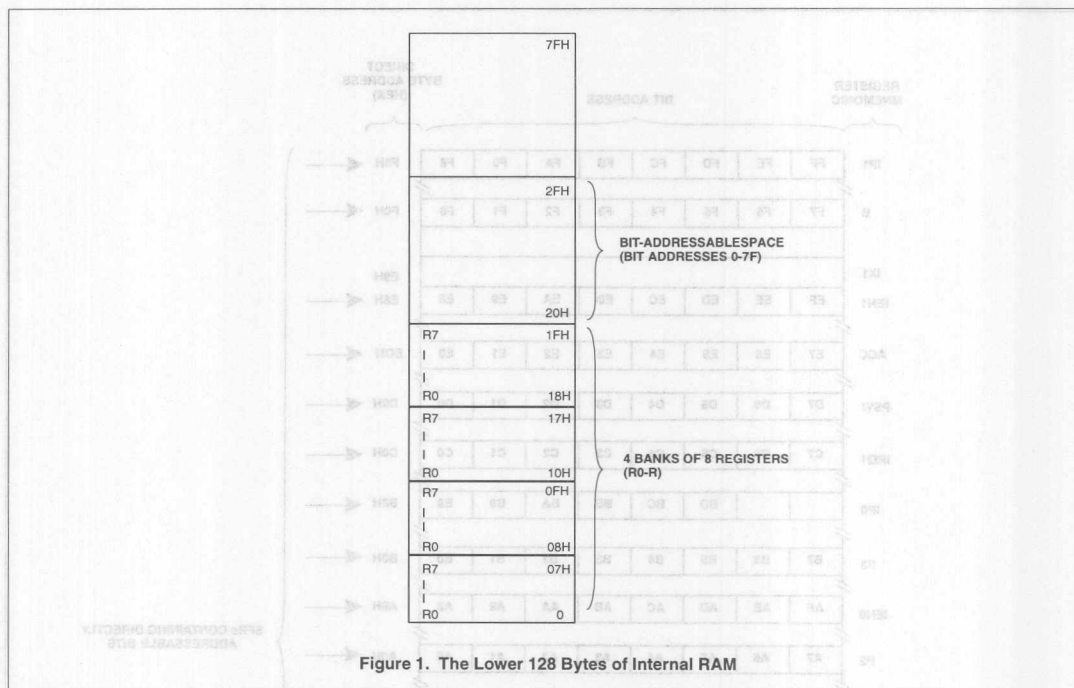
The 80CL51 has five methods for addressing source operands:

- Register
- Direct
- Register-Indirect
- Immediate
- Base-Register-plus Index-Register-indirect

## MEMORY MAP



## Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51



The first three methods can be used for addressing destination operands. Most instructions have a "destination/source" field that specifies data type, addressing methods and operands involved. For operations other than MOVs, the destination operand is also a source operand.

Access to memory addressing is as follows:

- Registers in one of the four register banks through register, direct or indirect.
- Internal RAM (128 bytes) through direct or register-indirect.
- Special Function Register through Direct.
- External data memory through Register-Indirect
- Program memory look-up tables through Base-Register-Plus Index-Register-Indirect.

## 1.2 I/O Facilities

### 1.2.1 Ports

The 80CL51 has 32 I/O lines treated as 32 individually addressable bits or as four parallel 8-bit addressable ports. Port 0, 1, 2 and 3 perform the following alternate functions:

- Port 0: provides the multiplexed low-order address and data bus for expanding the device with standard memories and peripherals.
- Port 1: provides the inputs for the external interrupts INT2/INT9.
- Port 2: provides the high-order address when expanding the device with external program or data memory.
- Port 3: pins can be configured individually to provide:
  - (1) external interrupt request inputs
  - (2) counter input
  - (3) control signals to read and write to external memories
  - (4) UART input and output

To enable a Port 3 pin alternate function, the Port 3 bit latch in its SFR must contain a logic 1.

Each port consists of a latch (Special Function Registers P0 to P3), an output driver and an input buffer. Ports 1,2,3 have internal pull ups. Figure 3(a) shows that the strong transistor p1 is turned on for only 2 oscillator periods after a 0-to-1 transition in

the port latch. When on, it turns on p3 (a weak pull up) through the inverter. This inverter and p3 form a latch which hold the 1. In Port 0 the pull up p1 is only on when emitting 1s for external memory access. Writing a 1 to a Port 0 bit latch leaves both output transistors switched off so the pin can be used as a high-impedance input.

### 1.2.2 Port Options

The pins of port 1, port 2, and port 3 may be individually configured with one of the following options (see Figure 3):

Option 1: Standard Port; quasi-bidirectional I/O with pull up. The strong booster pull up p1 is turned on for two oscillator periods after a 0-to-1 transition in the port latch (see Figure 3(a)).

Option 2: Open drain; quasi-bidirectional I/O with n-channel open drain output. Use as an output requires the connection of an external pull up resistor (see Figure 3(c)).

Option 3: Push-Pull; output with drive capability in both polarities. Under this option, pins can only be used as outputs. See Figure 3(b).

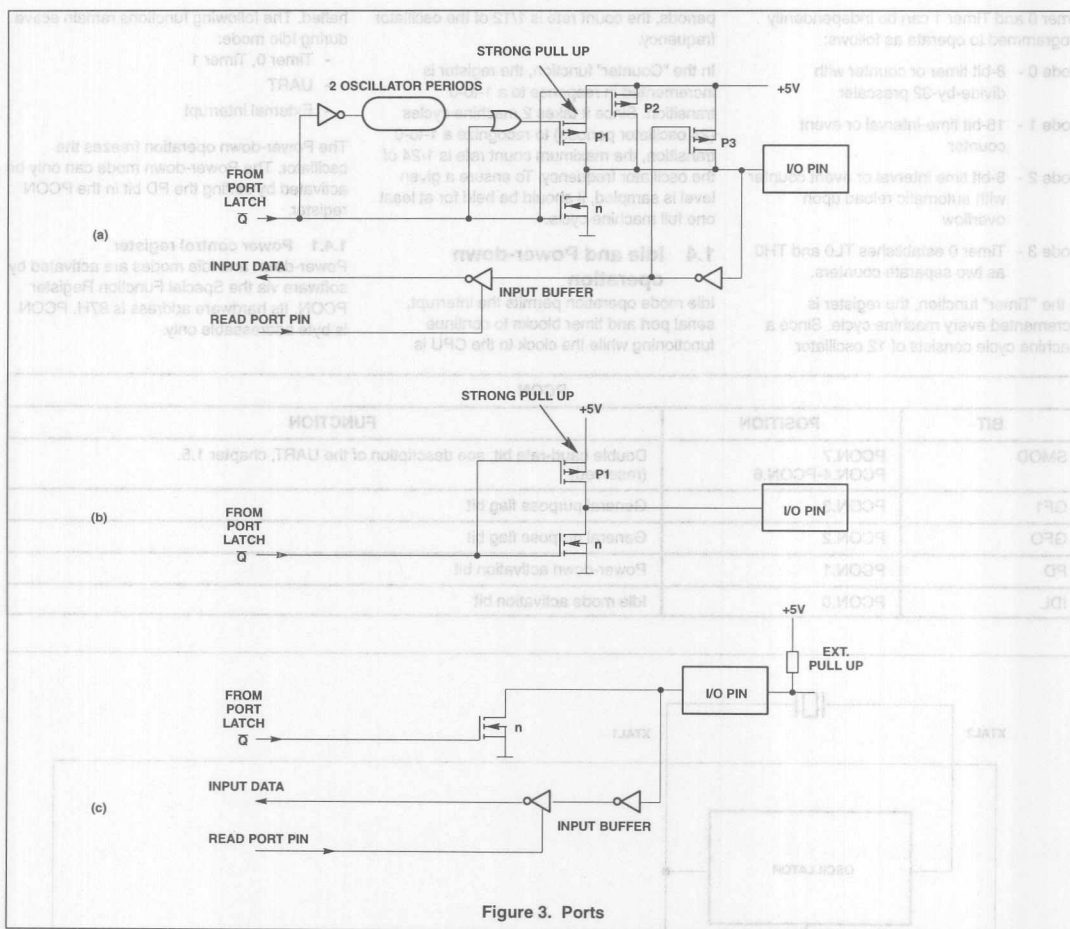
## Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

REGISTER MNEMONIC	BIT ADDRESS								DIRECT BYTE ADDRESS (HEX)
IP1	FF	FE	FD	FC	FB	FA	F9	F8	F8H
B	F7	F6	F5	F4	F3	F2	F1	F0	F0H
IX1									E9H
IEN1	EF	EE	ED	EC	EB	EA	E9	E8	E8H
ACC	E7	E6	E5	E4	E3	E2	E1	E0	E0H
PSW	D7	D6	D5	D4	D3	D2	D1	D0	D0H
IRQ1	C7	C6	C5	C4	C3	C2	C1	C0	C0H
IP0									B8H
P3	B7	B6	B5	B4	B3	B2	B1	B0	B0H
IEN0	AF	AE	AD	AC	AB	AA	A9	A8	A8H
P2	A7	A6	A5	A4	A3	A2	A1	A0	A0H
S0BUF									99H
S0CON	9F	9E	9D	9C	9B	9A	99	98	98H
P1	97	96	95	94	93	92	91	90	90H
TH1									8DH
TH0									8CH
TL1									8BH
TL0									8AH
TMOD									89H
TCON	8F	8E	8D	8C	8B	8A	89	88	88H
PCON									87H
DPH									83H
DPL									82H
SP									81H
P0	87	86	85	84	83	82	81	80	80H

SFRs CONTAINING DIRECTLY  
ADDRESSABLE BITS

Figure 2. Special Function Registers

## Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51



The definition of port options for port 0 is slightly different. Two cases have to be examined. First, accesses to external memory (EA=0 or access above the built-in memory boundary), second, I/O accesses.

#### External Memory Accesses

- Option 1: True 0 and 1 are written as address to the external memory (strong pull up is used).
- Option 2: An external pull up resistor is needed for external accesses.
- Option 3: Not allowed for external memory access as the port can only be used as output.

#### I/O Accesses

- Option 1: When writing a 1 to the port-latch, the strong pull up p1 will be on for 2 oscillator periods. No weak pull up exists. Without an external pull up, this option can be used as a high-impedance input.
- Option 2: Open drain; quasi-bidirectional I/O with n-channel open drain output. Use as an output requires the connection of an external pull up resistor (see Figure 3(c)).
- Option 3: Push-Pull; output with drive capability in both polarities. Under this option, pins can only be used as outputs.

Individual mask selection of the post-reset state is available on any of the above pins. Make your selection by appending "S" or "R" to option 1, 2, or 3 above (e.g. 1 S for a standard I/O to be set after RESET or 2 R for an open-drain I/O to be reset after RESET).

#### 1.3 Timer/event counter

The 80CL51 contains two 16-bit Timer/Counter registers, Timer 0 and Timer 1, which can perform the following functions:

- Measure time intervals and pulse durations
- Count events
- Generate interrupts requests

## Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

Timer 0 and Timer 1 can be independently programmed to operate as follows:

- Mode 0 - 8-bit timer or counter with divide-by-32 prescaler
- Mode 1 - 16-bit time-interval or event counter
- Mode 2 - 8-bit time interval or event counter with automatic reload upon overflow
- Mode 3 - Timer 0 establishes TL0 and TH0 as two separate counters.

In the "Timer" function, the register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator

periods, the count rate is 1/12 of the oscillator frequency.

In the "Counter" function, the register is incremented in response to a 1-to-0 transition. Since it takes 2 machine cycles (24 oscillator periods) to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. To ensure a given level is sampled, it should be held for at least one full machine cycle.

### 1.4 Idle and Power-down operation

Idle mode operation permits the interrupt, serial port and timer blocks to continue functioning while the clock to the CPU is

halted. The following functions remain active during Idle mode:

- Timer 0, Timer 1
- UART
- External interrupt

The Power-down operation freezes the oscillator. The Power-down mode can only be activated by setting the PD bit in the PCON register.

#### 1.4.1 Power control register

Power-down and Idle modes are activated by software via the Special Function Register PCON. Its hardware address is 87H. PCON is byte addressable only.

PCON		
BIT	POSITION	FUNCTION
SMOD	PCON.7 PCON.4-PCON.6	Double baud-rate bit, see description of the UART, chapter 1.5. (reserved)
GF1	PCON.3	General purpose flag bit
GFO	PCON.2	General purpose flag bit
PD	PCON.1	Power-down activation bit
IDL	PCON.0	Idle mode activation bit

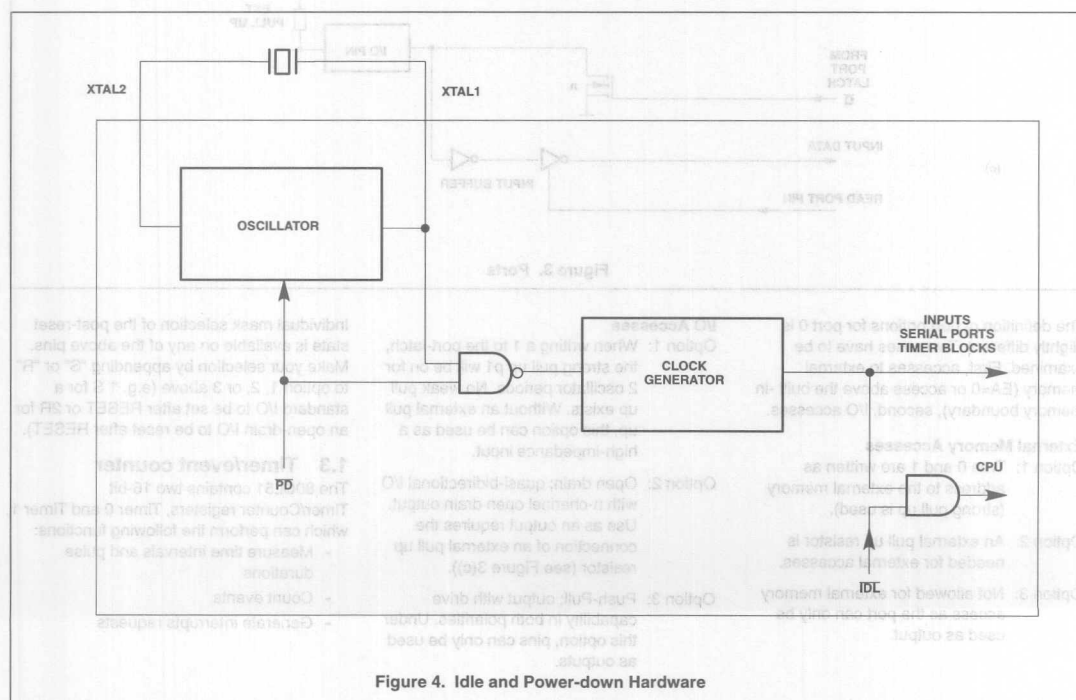


Figure 4. Idle and Power-down Hardware



## Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

### 1.4.2 Power-down mode

The instruction setting PCON.1 is the last executed prior to going into the Power-down mode. In Power-down mode the oscillator is stopped. The contents of the on-chip RAM and SFRs are preserved. The port pins output the values held by their respective SFRs. ALE and PSEN are held LOW.

In the Power-down mode  $V_{DD}$  may be reduced to minimize power consumption. However, the supply voltage must not be reduced until Power-down mode is active, and must be restored before the hardware reset is applied and frees the oscillator. Reset must be held active until the oscillator has restarted and stabilized.

The wake-up operation after power-down in this controller has two basic approaches:

#### 1.4.2.1 Wake-up using INT2 to INT9

If INT2 to INT9 are enabled, the 80CL51 can be awakened from power-down mode with the external interrupts. To ensure that the oscillator is stable before the controller restarts, the internal clock will remain inactive for 1536 oscillator periods. This is controlled by an on-chip delay counter.

### 1.4.2.2 Wake-up using RESET

To wake-up the 80CL51 the RESET pin has to be kept HIGH for a minimum of 24 oscillator periods. The on-chip delay counter is inactive. The user has to ensure that the oscillator is stable before any operation is attempted. Figure 5 illustrates the two possibilities for wake-up.

### 1.4.3 Idle mode

The instruction that sets PCON.0 is the last instruction executed before going into Idle mode. Once in the Idle mode, the internal clock is gated away from the CPU, but not from the Interrupt, Timer and Serial port functions. The CPU status is preserved along with the Stack Pointer, Program Counter, Program Status Word and Accumulator. The RAM and all other registers maintain their data during Idle mode. The port pins retain the logical states they held at Idle mode activation. ALE and PSEN hold at the logic HIGH level.

There are two methods used to terminate the Idle mode. Activation of any enabled interrupt will cause PCON to be cleared by hardware, terminating Idle mode. The interrupt is serviced, and following the instruction RETI, the next instruction to be executed will be the

one following the instruction that put the device in the Idle mode.

Flag bits GF0 and GF1 may be used to determine whether the interrupt was received during normal execution or Idle mode. For example, the instruction that writes to PCON.0 can also set or clear one or both flag bits. When Idle mode is terminated by an interrupt, the service routine can examine the status of the flag bits.

The second method of terminating the Idle mode is with an external hardware reset. Since the oscillator is still running, the hardware reset is required to be active for only two machine cycles to complete the reset operation.

Reset redefines all SFRs, but does not affect the on-chip RAM.

The status of the external pins during Idle and Power-down mode is shown in Table 4. If the Power-down mode is activated while accessing external memory, port data held in the Special Function Register P2 is restored to Port 2. If the data is a logic 1, the port pin is held HIGH during the Power-down mode by the strong pull up transistor p1 (see Figure 3(a)).

Table 4. Status of the External Pins During Idle and Power-down Mode

MODE	MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle	internal	1	1	Port Data	Port Data	Port Data	Port Data
Idle	external	1	1	Floating	Port Data	Address	Port Data
Power-down	internal	0	0	Port Data	Port Data	Port Data	Port Data
Power-down	external	0	0	Floating	Port Data	Port Data	Port Data

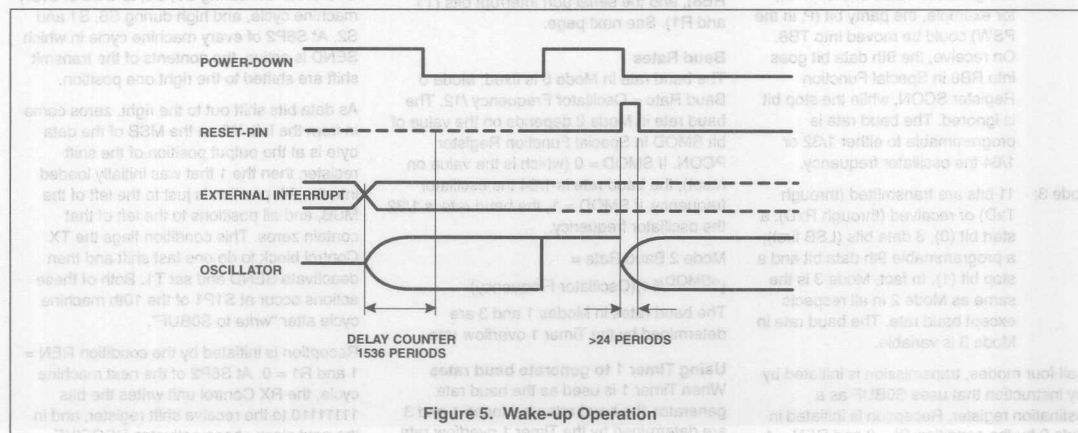


Figure 5. Wake-up Operation



## Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

### 1.5 Standard serial interface SIO: UART

This serial port is full duplex, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the register. (However, if the first byte still hasn't been read by the time reception of the second byte is complete, one of the bytes will be lost). The serial port receive and transmit registers are both accessed at Special Function Register S0BUF. Writing to S0BUF loads the transmit register, and reading S0BUF loads the transmit register, and reading S0BUF accesses a physically separate receive register.

The serial port can operate in 4 modes:

- Mode 0:** Serial data enters and exits through RxD. TxD outputs the shift clock. 8 bits are transmitted/received (LSB first). The baud is fixed at 1/12 the oscillator frequency.
- Mode 1:** 10 bits are transmitted (through TxD) or received (through RxD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in Special Function Register SCON. The baud rate is variable.
- Mode 2:** 11 bits are transmitted (through TxD) or received (through RxD): start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On Transmit, the 9th data bit (TB8 in SCON) can be assigned the value of 0 or 1. Or, for example, the parity bit (P, in the PSW) could be moved into TB8. On receive, the 9th data bit goes into RB8 in Special Function Register SCON, while the stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 the oscillator frequency.
- Mode 3:** 11 bits are transmitted (through TxD) or received (through RxD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit and a stop bit (1). In fact, Mode 3 is the same as Mode 2 in all respects except baud rate. The baud rate in Mode 3 is variable.

In all four modes, transmission is initiated by any instruction that uses S0BUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit if REN = 1.

#### 1.5.1 Multiprocessor communications

Modes 2 and 3 have a special provision for multiprocessor communications. In these modes, 9 data bits are received. The 9th one goes into RB8. Then comes a stop bit. The port can be programmed such that when the stop bit is received, the serial port interrupt will be activated only if RB8 = 1. This feature is enabled by setting bit SM2 in SCON. A way to use this feature in multiprocessor systems is as follows:

When the master processor wants to transmit a block of data to one of several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte. With SM2 = 1, no slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave will clear its SM2 bit and prepare to receive the data bytes that will be coming. The slaves that weren't being addressed leave their SM2s set and go on about their business, ignoring the coming data bytes.

SM2 has no effect in Mode 0, and in Mode 1 can be used to check the validity of the stop bit. In a Mode 1 reception, if SM2 = 1, the receive interrupt will not be activated unless a valid stop bit is received.

#### 1.5.2 Serial port control register

The serial port control and status register is the Special Function Register S0CON, shown in Figure 6. The register contains not only the mode selection bits, but also the 9th data bit for transmit and receive (TB8 and RB8), and the serial port interrupt bits (T1 and R1). See next page.

#### Baud Rates

The baud rate in Mode 0 is fixed: Mode 0 Baud Rate = Oscillator Frequency / 12. The baud rate in Mode 2 depends on the value of bit SMOD in Special Function Register PCON. If SMOD = 0 (which is the value on reset), the baud rate is 1/64 the oscillator frequency. If SMOD = 1, the baud rate is 1/32 the oscillator frequency.

Mode 2 Baud Rate =  $(2^{SMOD}/64)(\text{Oscillator Frequency})$

The baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate.

#### Using Timer 1 to generate baud rates

When Timer 1 is used as the baud rate generator, the baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate and the value of SMOD as follows:

$(2^{SMOD}/32)(\text{Timer 1 Overflow Rate})$

The Timer 1 interrupt should be disabled in this application. The Timer itself can be configured for either "timer" or "counter" operation, and in any of its 3 running modes. In the most typical applications, it is configured for "timer operation, in the auto-reload mode (high nibble of TMOD = 0010B). In that case the baud rate is given by the formula:

Mode 1, 3 Baud Rate =  $\{ (2^{SMOD}/32) (\text{Oscillator Frequency}) \} / \{ 12 (256 - (TH 1)) \}$

One can achieve very low baud rates with Timer 1 by leaving the Timer 1 interrupt enabled, and configuring this Timer to run as a 16-bit timer (high nibble of TMOD = 0001B), and using the Timer 1 interrupt to do a 16-bit software reload. Table 5 lists various commonly used baud rates and how they can be obtained from Timer 1.

#### More about Mode 0

Figure 7 shows a simplified functional diagram of the serial port in Mode 0, and associated timing. Transmission is initiated by any instruction that uses S0BUF as a destination register. The "write to S0BUF" signal at S6P2 also loads a 1 into the 9th position of the transmit shift register and tells the TX Control block to commence a transmission. The internal timing is such that the one full machine cycle will elapse between "write to S0BUF", and activation of SEND.

SEND enables the output of the shift register to the alternate output function line of P3.0 and also enables SHIFT CLOCK to the alternate output function line of P3.1. SHIFT CLOCK is low during S3, S4, and S5 of every machine cycle, and high during S6, S1 and S2. At S6P2 of every machine cycle in which SEND is active, the contents of the transmit shift are shifted to the right one position.

As data bits shift out to the right, zeros come in from the left. When the MSB of the data byte is at the output position of the shift register, then the 1 that was initially loaded into the 9th position is just to the left of the MSB, and all positions to the left of that contain zeros. This condition flags the TX Control block to do one last shift and then deactivate SEND and set T1. Both of these actions occur at S1P1 of the 10th machine cycle after "write to S0BUF".

Reception is initiated by the condition REN = 1 and R1 = 0. At S6P2 of the next machine cycle, the RX Control unit writes the bits 11111110 to the receive shift register, and in the next clock phase activates RECEIVE.

## Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

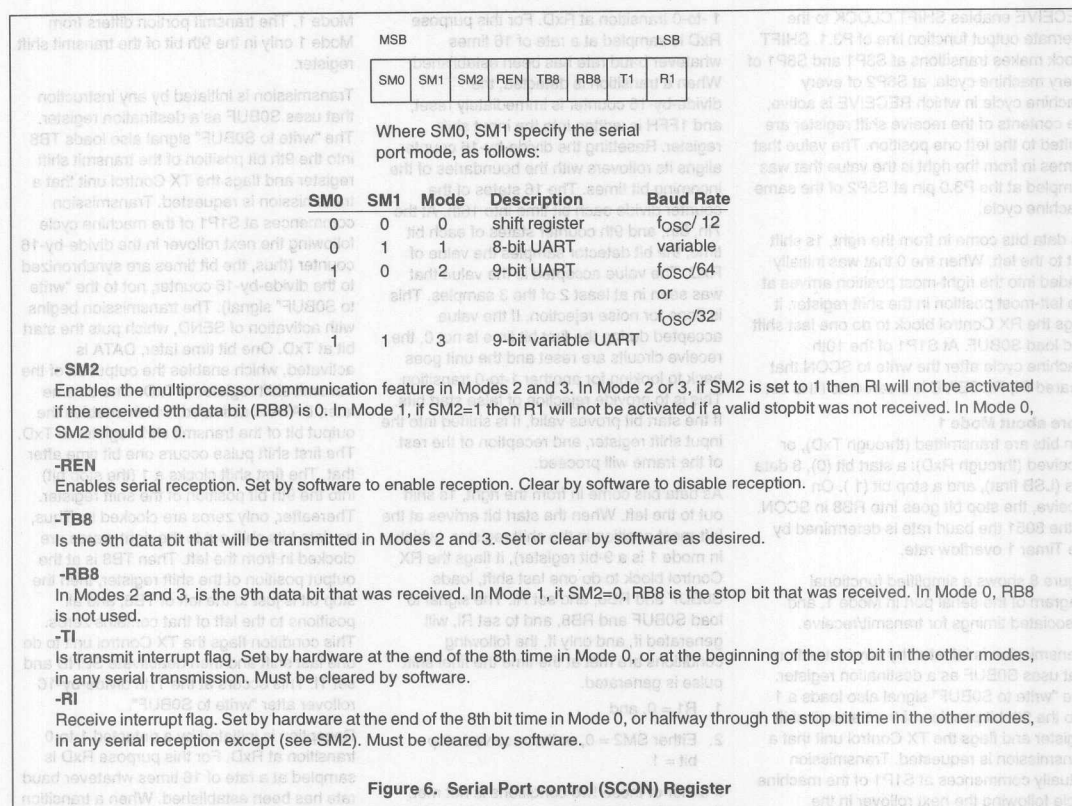


Table 5. Timer 1 Generated Commonly Used Baud Rates

TIMER 1					
BAUD RATE	$f_{osc}$	SMOD	C/T	MODE	RELOAD VALUE
Mode 0 Max: 1.33 Mb/s	16 MHz	x	x	x	x
Mode 2 Max: 500 Kb/s	16 MHz	1	x	x	x
Modes 1,3: 83.3 Kb/s	16 MHz	1	0	2	FFH
19.2 Kb/s	11.059 MHz	1	0	2	FDH
9.6 Kb/s	11.059 MHz	0	0	2	FDH
4.8 Kb/s	11.059 MHz	0	0	2	FAH
2.4 Kb/s	11.059 MHz	0	0	2	F4H
1.2 Kb/s	11.059 MHz	0	0	2	E8H
137.5 Kb/s	11.986 MHz	0	0	2	1DH
110	6 MHz	0	0	2	72H
110	12 MHz	0	0	1	FEEDH

## Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

RECEIVE enables SHIFT CLOCK to the alternate output function line of P3.1. SHIFT Clock makes transitions at S3P1 and S6P1 of every machine cycle. At S6P2 of every machine cycle in which RECEIVE is active, the contents of the receive shift register are shifted to the left one position. The value that comes in from the right is the value that was sampled at the P3.0 pin at S5P2 of the same machine cycle.

As data bits come in from the right, 1s shift out to the left. When the 0 that was initially loaded into the right-most position arrives at the left-most position in the shift register, it flags the RX Control block to do one last shift and load S0BUF. At S1P1 of the 10th machine cycle after the write to SCON that cleared RI, RECEIVE is cleared as RI is set.

### More about Mode 1

Ten bits are transmitted (through TxD), or received (through RxD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in SCON. In the 8051 the baud rate is determined by the Timer 1 overflow rate.

Figure 8 shows a simplified functional diagram of the serial port in Mode 1, and associated timings for transmit/receive.

Transmission is initiated by any instruction that uses S0BUF as a destination register. The "write to S0BUF" signal also loads a 1 into the 9th bit position of the transmit shift register and flags the TX Control unit that a transmission is requested. Transmission actually commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter. (Thus, the bit times are synchronized to the divide-by-16 counter, not to the "write to S0BUF" signal).

The transmission begins with activation of SEND which sends the start bit to pin TxD. One bit time later, DATA is activated, enabling the transmission of the output bit of the transmit shift register to TxD. The first shift pulse occurs one bit time after that.

As data bits shift out to the right, zeros are clocked in from the left. When the MSB of the data byte is at the output position of the shift register, then the 1 that was initially loaded into the 9th position is just to the left of the MSB, and all positions to the left of that contain zeros. This condition flags the TX Control unit to do one last shift and then deactivate SEND and set TI. This occurs at the 10th divide-by-16 rollover after "write to S0BUF". Reception is initiated by a detected

1-to-0 transition at RxD. For this purpose RxD is sampled at a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FFH is written into the input shift register. Resetting the divide-by-16 counter aligns its rollovers with the boundaries of the incoming bit times. The 16 states of the counter divide each bit time into 16th. At the 7th, 8th, and 9th counter states of each bit time, the bit detector samples the value of RxD. The value accepted is the value that was seen in at least 2 of the 3 samples. This is done for noise rejection. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back to looking for another 1-to-0 transition. This is to provide rejection of false start bits. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

As data bits come in from the right, 1s shift out to the left. When the start bit arrives at the left-most position in the shift register, (which in mode 1 is a 9-bit register), it flags the RX Control block to do one last shift, loads S0BUF and RB8, and set RI. The signal to load S0BUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated.

1. R1 = 0, and
2. Either SM2 = 0, or the received stop bit = 1

If either of these two conditions is not met, the received frame is irretrievably lost. If both conditions are met, the stop bit goes into RB8, the 8 data bits go into S0BUF, and RI is activated. At this time, whether the above conditions are met or not, the unit goes back to looking for a 1-to-0 transition in RxD.

### More about modes 2 and 3

Eleven bits are transmitted (through TxD), or received (through RxD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On transmit, the 9th data bit (TB8) can be assigned the value of 0 or 1. On receive, the 9th data bit goes into RB8 in SCON. The baud rate is programmable to either 1/32 or 1/64 the oscillator frequency in Mode 2. Mode 3 may have a variable baud rate generated from Timer 1.

Figures 9 and 10 show a functional diagram of the serial port in Modes 2 and 3. The receive portion is exactly the same as in

Mode 1. The transmit portion differs from Mode 1 only in the 9th bit of the transmit shift register.

Transmission is initiated by any instruction that uses S0BUF as a destination register. The "write to S0BUF" signal also loads TB8 into the 9th bit position of the transmit shift register and flags the TX Control unit that a transmission is requested. Transmission commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter (thus, the bit times are synchronized to the divide-by-16 counter, not to the "write to S0BUF" signal). The transmission begins with activation of SEND, which puts the start bit at TxD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TxD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TxD. The first shift pulse occurs one bit time after that. The first shift clocks a 1 (the stop bit) into the 9th bit position of the shift register. Thereafter, only zeros are clocked in. Thus, as data bits shift out to the right, zeros are clocked in from the left. Then TB8 is at the output position of the shift register, then the stop bit is just to the left of TB8, and all positions to the left of that contain zeros. This condition flags the TX Control unit to do one last shift and then deactivate SEND and set TI. This occurs at the 11th divide-by-16 rollover after "write to S0BUF".

Reception is initiated by a detected 1-to-0 transition at RxD. For this purpose RxD is sampled at a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FFH is written to the input shift register.

At the 7th, 8th and 9th counter states of each bit time, the bit detector samples the value of RxD. The value accepted is the value that was seen in at least 2 of the 3 samples. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back to looking for another 1-to-0 transition. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed. As data bits come in from the right, 1s shift out to the left. When the start bit arrives at the left-most position in the shift register (which in Modes 2 and 3 is a 9-bit register), it flags the RX Control block to do one last shift, load S0BUF and RB8, and set RI.

## Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

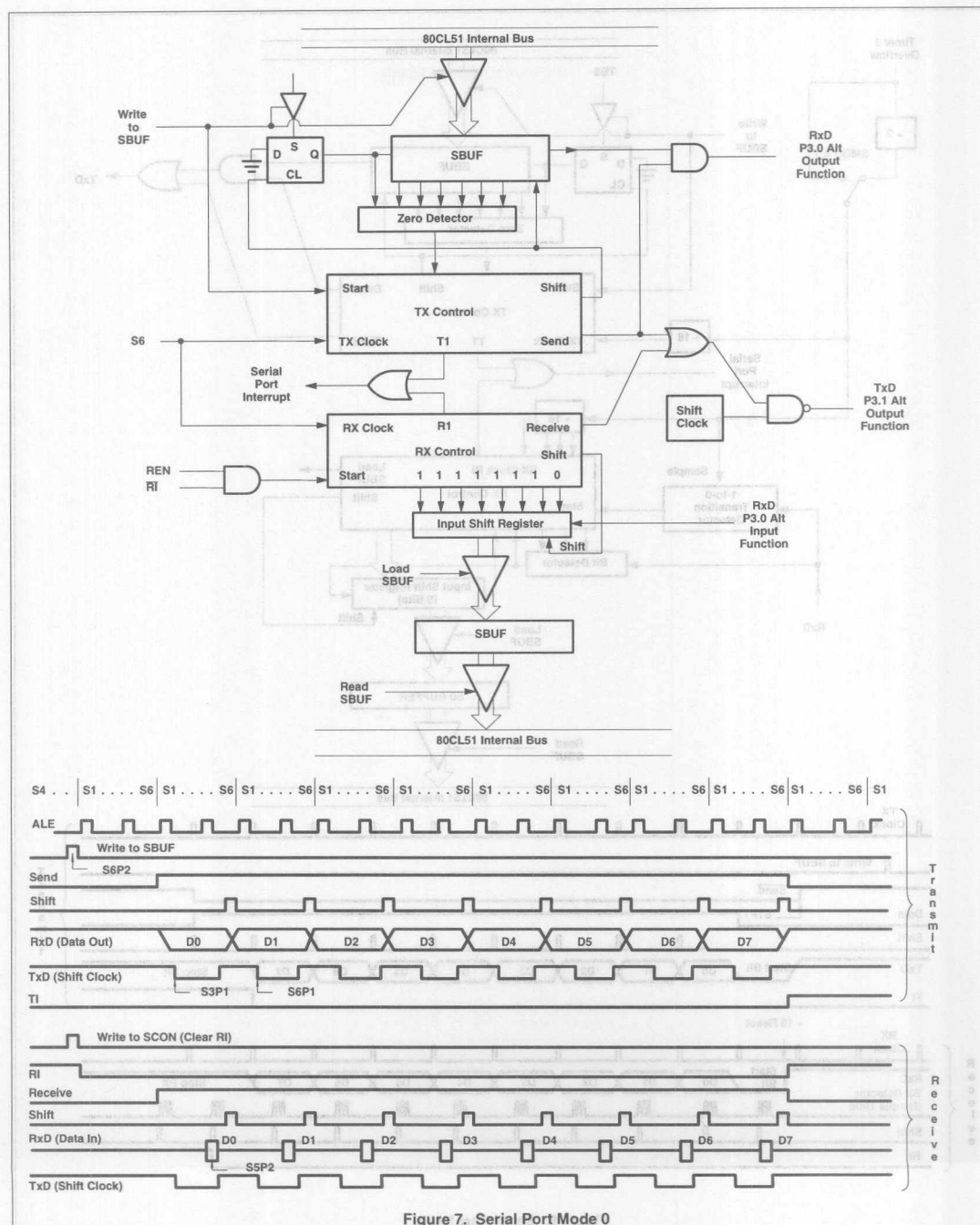


Figure 7. Serial Port Mode 0

## Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

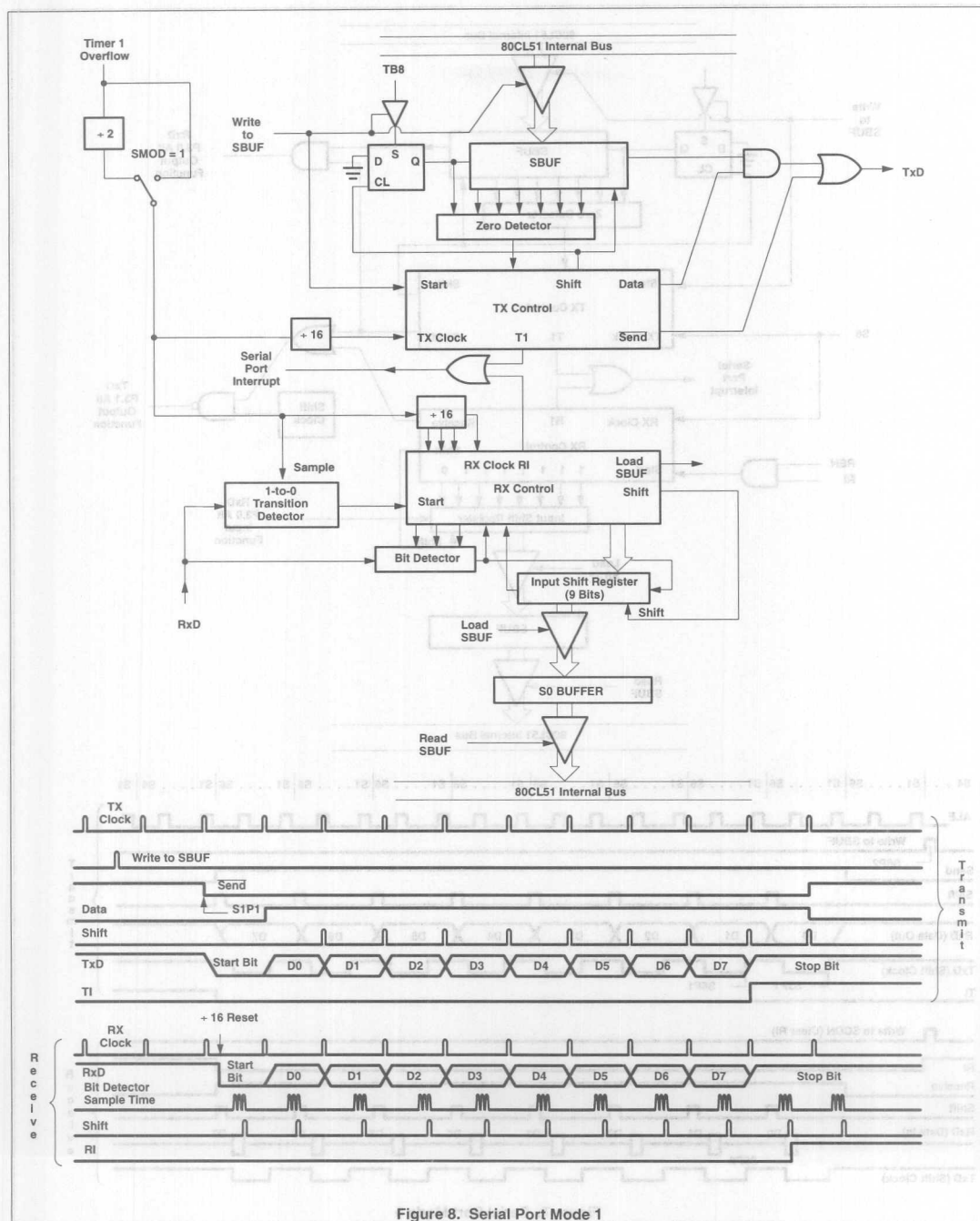


Figure 8. Serial Port Mode 1



## Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

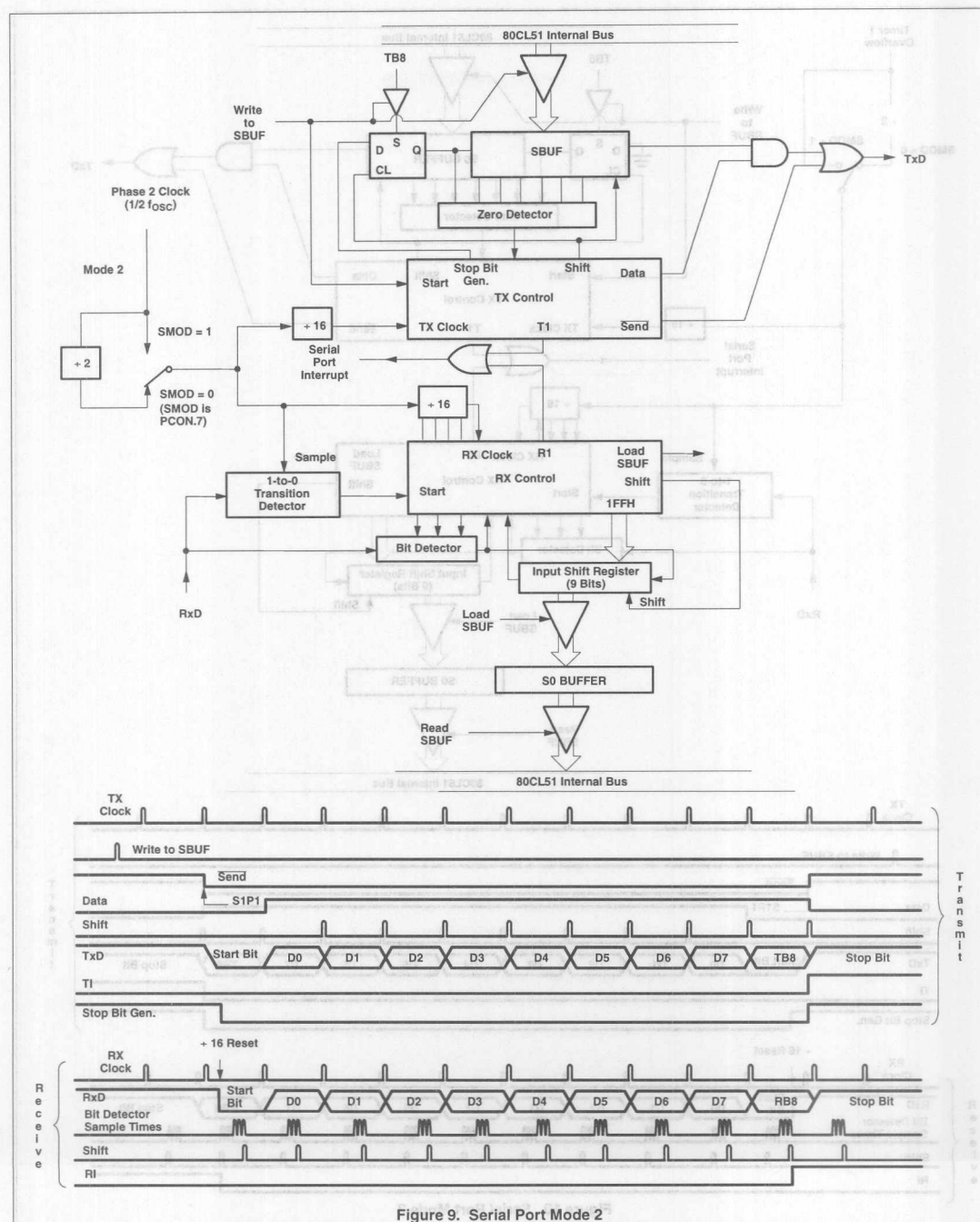


Figure 9. Serial Port Mode 2



## Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

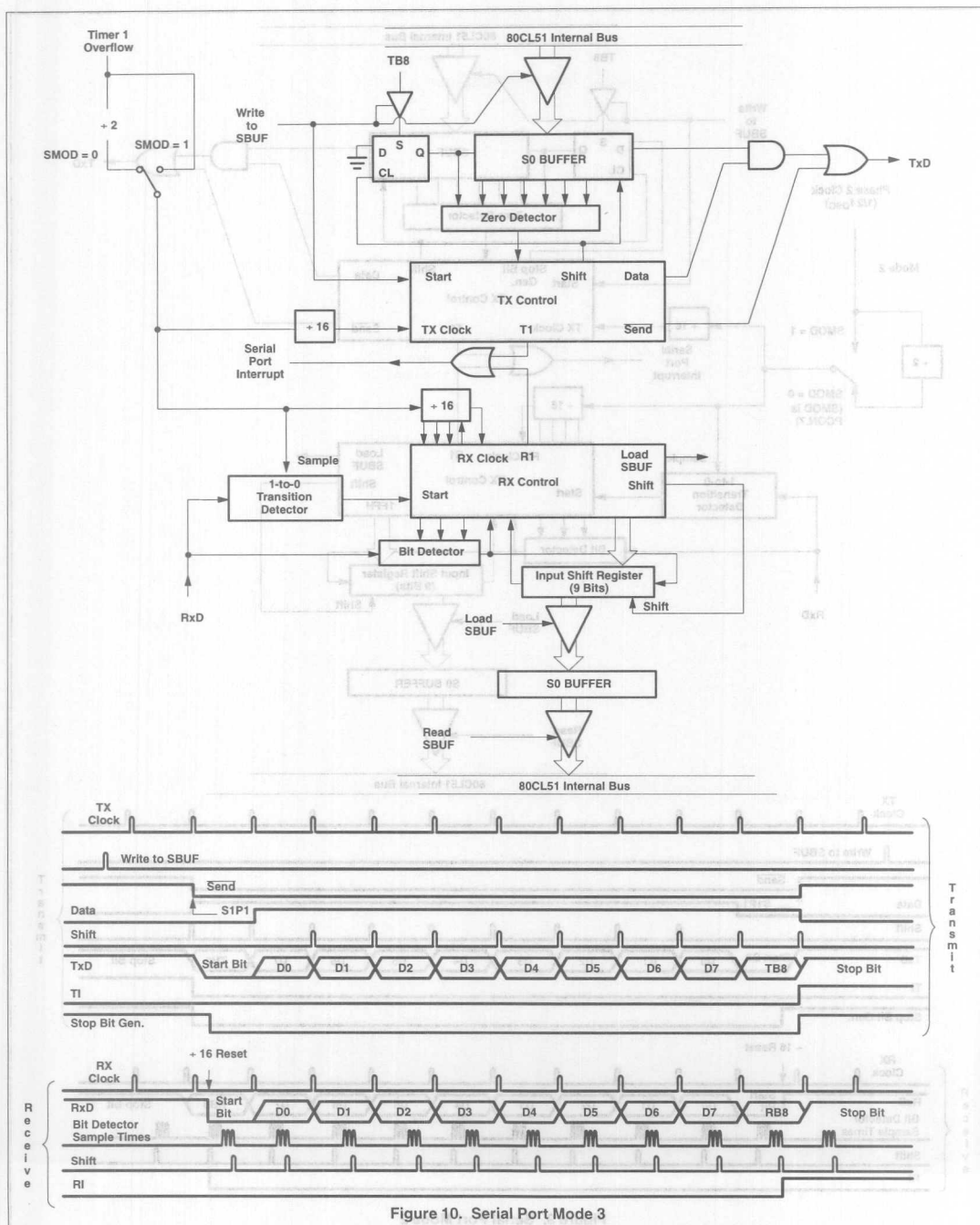


Figure 10. Serial Port Mode 3

## Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

The signal to load S0BUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated.

1. RI = 0, and
2. Either SM2 = 0 or the received 9th data bit = 1

If either of these conditions is not met, the received frame is irretrievably lost, and RI is not set. If both conditions are met, the received 9th data bit goes into RB8, and the first 8 data bits go into S0BUF. One bit time later, whether the above conditions were met or not, the unit goes back to looking for a 1-to-0 transition at the RxD input.

### 1.6 Interrupt System

External events and the real-time-driven on-chip peripherals require service by the CPU asynchronous to do execution of any

particular section of code. To tie the asynchronous activities of these functions to normal program execution, a multiple-source, two-priority-level, nested interrupt system is provided. The 80CL51 acknowledges interrupt requests from thirteen sources as follows:

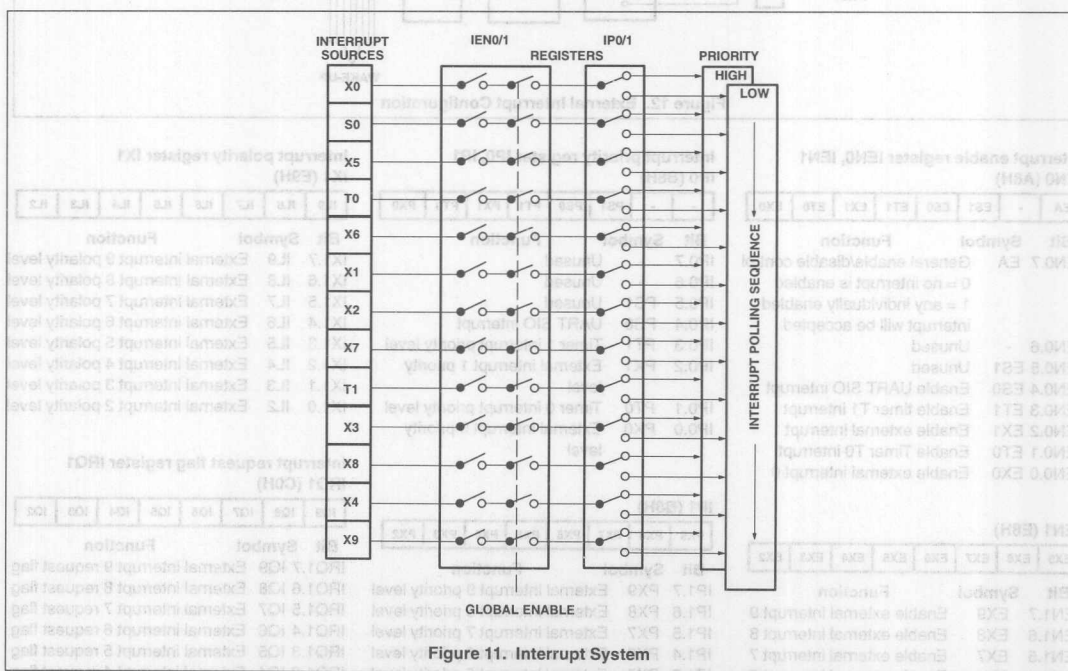
- INT0 and INT1
- Timer 0 and Timer 1
- UART serial I/O
- INT2 to INT9 (Port 1)

Each interrupt vectors to a separate location in program memory for its service routine. Each source can be individually enabled or disabled by corresponding bits in the Interrupt Enable Registers (IE, IEO). The priority level is selected via the Interrupt Priority register (IP0, IP1). All enabled sources can be globally disabled or enabled.

#### 1.6.1 External Interrupts INT2/INT9

Port 1 lines serve an alternative purpose as eight additional interrupts INT2 to INT9. When enabled, each of these lines may "wake-up" the device from Power-down mode. Using the IX1 register, each pin may be initialized to either active HIGH or LOW. IRQ1 is the interrupt request flag register. Each flag, if the interrupt is enabled, will be set on an interrupt request but must be cleared by software, i.e. via the interrupt software or when the interrupt is disabled.

The Port 1 interrupts are level sensitive. A Port 1 interrupt will be recognized when a level (HIGH or LOW) depending on Interrupt Polarity Register (IX1) on P1x is held active for at least one machine cycle. The Interrupt Request is not served until the next machine cycle.



## Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

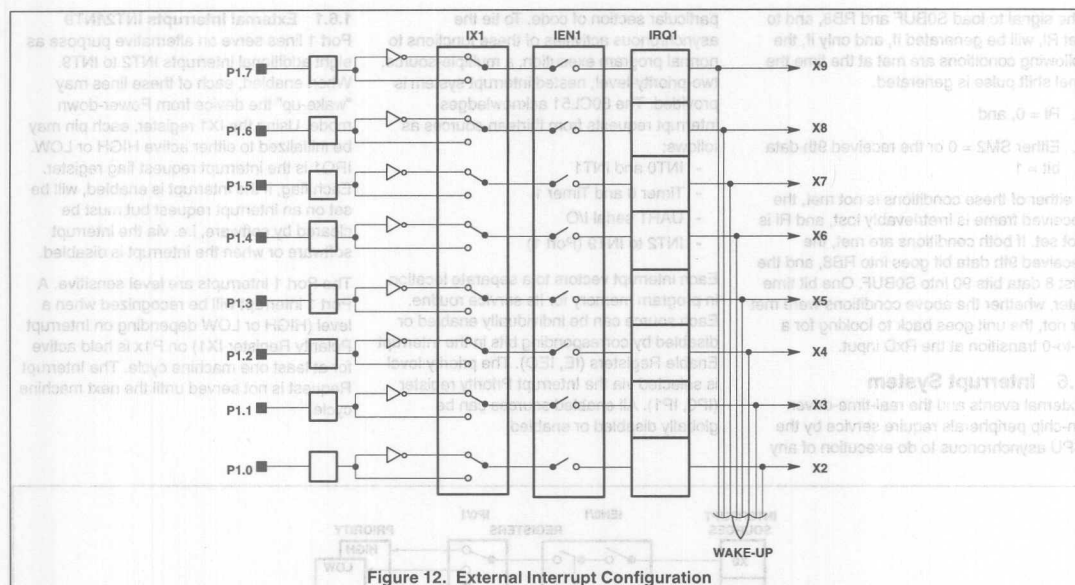


Figure 12. External Interrupt Configuration

Interrupt enable register IEN0, IEN1  
IEN0 (A8H)

EA	-	ES1	ES0	ET1	EX1	ET0	EX0
----	---	-----	-----	-----	-----	-----	-----

Bit	Symbol	Function
IEN0.7	EA	General enable/disable control 0 = no interrupt is enabled 1 = any individually enabled interrupt will be accepted
IEN0.6	-	Unused
IEN0.5	ES1	Unused
IEN0.4	ES0	Enable UART SIO interrupt
IEN0.3	ET1	Enable timer T1 interrupt
IEN0.2	EX1	Enable external interrupt
IEN0.1	ET0	Enable Timer T0 interrupt
IEN0.0	EX0	Enable external interrupt 0

## IEN1 (E8H)

EX9	EX8	EX7	EX6	EX5	EX4	EX3	EX2
-----	-----	-----	-----	-----	-----	-----	-----

Bit	Symbol	Function
IEN1.7	EX9	Enable external interrupt 9
IEN1.6	EX8	Enable external interrupt 8
IEN1.5	EX7	Enable external interrupt 7
IEN1.4	EX6	Enable external interrupt 6
IEN1.3	EX5	Enable external interrupt 5
IEN1.2	EX4	Enable external interrupt 4
IEN1.1	EX3	Enable external interrupt 3
IEN1.0	EX2	Enable external interrupt 2

where 0 = interrupt disabled  
1 = interrupt enabled

Interrupt priority register IP0, IP1  
IP0 (B8H)

-	-	PS1	PS0	PT1	PX1	PT0	PX0
---	---	-----	-----	-----	-----	-----	-----

Bit	Symbol	Function
IP0.7	-	Unused
IP0.6	-	Unused
IP0.5	PS1	Unused
IP0.4	PS0	UART SIO interrupt
IP0.3	PT1	Timer 1 interrupt priority level
IP0.2	PX1	External interrupt 1 priority level
IP0.1	PT0	Timer 0 interrupt priority level
IP0.0	PX0	External interrupt 0 priority level

## IP1 (B8H)

PX9	PX8	PX7	PX6	PX5	PX4	PX3	PX2
-----	-----	-----	-----	-----	-----	-----	-----

Bit	Symbol	Function
IP1.7	PX9	External interrupt 9 priority level
IP1.6	PX8	External interrupt 8 priority level
IP1.5	PX7	External interrupt 7 priority level
IP1.4	PX6	External interrupt 6 priority level
IP1.3	PX5	External interrupt 5 priority level
IP1.2	PX4	External interrupt 4 priority level
IP1.1	PX3	External interrupt 3 priority level
IP1.0	PX2	External interrupt 2 priority level

Interrupt priority is as follows:  
0 = low priority  
1 = high priority

Interrupt polarity register IX1  
IX1 (E9H)

IL9	IL8	IL7	IL6	IL5	IL4	IL3	IL2
-----	-----	-----	-----	-----	-----	-----	-----

Bit	Symbol	Function
IX1.7	IL9	External interrupt 9 polarity level
IX1.6	IL8	External interrupt 8 polarity level
IX1.5	IL7	External interrupt 7 polarity level
IX1.4	IL6	External interrupt 6 polarity level
IX1.3	IL5	External interrupt 5 polarity level
IX1.2	IL4	External interrupt 4 polarity level
IX1.1	IL3	External interrupt 3 polarity level
IX1.0	IL2	External interrupt 2 polarity level

Interrupt request flag register IRQ1  
IRQ1 (C0H)

IQ9	IQ8	IQ7	IQ6	IQ5	IQ4	IQ3	IQ2
-----	-----	-----	-----	-----	-----	-----	-----

Bit	Symbol	Function
IRQ1.7	IQ9	External interrupt 9 request flag
IRQ1.6	IQ8	External interrupt 8 request flag
IRQ1.5	IQ7	External interrupt 7 request flag
IRQ1.4	IQ6	External interrupt 6 request flag
IRQ1.3	IQ5	External interrupt 5 request flag
IRQ1.2	IQ4	External interrupt 4 request flag
IRQ1.1	IQ3	External interrupt 3 request flag
IRQ1.0	IQ2	External interrupt 2 request flag

# Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

## 1.6.2 Interrupt Vectors

	Vector	Source
X0	0003H	External 0
S0	0023H	UART SIO
X5	0053H	External 5
T0	000BH	Timer 0
X6	005BH	External 6
X1	0013H	External 1
X2	003BH	External 2
X7	0063H	External 7
T1	001BH	Timer 1
X3	0043H	External 3
X8	006BH	External 8
X4	004BH	External 4
X9	0073H	External 9

### Interrupt priority

Each interrupt priority source can be set to either high or low priority. If both priorities are requested simultaneously, the controller will branch to the high priority vector.

A low priority interrupt can only be interrupted by a high priority interrupt. A high priority interrupt routine cannot be interrupted.

## 1.6.3 Related registers

The following registers are used in conjunction with the interrupt system:

Register	Function
IX1	Interrupt polarity register
IRQ1	Interrupt enable register
IEN1	Interrupt enable register (INT2-INT9)
IP0	Interrupt priority register
IP1	Interrupt priority register (INT2-INT9)

## 1.7 Oscillator registers

The on-chip circuitry of the 80CL51 is a single-stage inverting amplifier biased by an internal feedback resistor (Figure 13). For operation as standard quartz oscillator, no external components are needed except at 32 KHz. When using external capacitors, ceramic resonators, coils and RC networks to drive the oscillator, five different configurations are supported (see Figure 14 and oscillator options).

In the Power-down mode the oscillator is stopped XTAL1 is pulled HIGH. The oscillator inverter is switched off to ensure no current will flow regardless of the voltage at XTAL1. To drive the device with an external clock source, apply the external clock signal to XTAL1, and leave XTAL2 to float, as shown

in Figure 14(f). There are no requirements on the duty cycle of the external clock, since the input to the internal clocking circuitry is split using a flip-flop.

The following options are provided for optimum on-chip oscillator performance. Please state option when ordering.

### 1.7.1 Oscillator options (see Figure 14)

The following options are provided for optimum on-chip oscillator performance. Please state option when ordering.

Osc. 1: : Figure 14(c): An option for 32 kHz clock applications with external trimmer for frequency adjustment. A 4.7 MΩ bias resistor is needed for use in parallel with the crystal.

Osc. 2: Figure 14(e): An option for low-power, low-frequency operations using LC components.

Osc. 3: An option for medium frequency range applications.

Osc. 4: An option for high frequency range applications.

RC: Figure 14(g): An option for an RC oscillator.

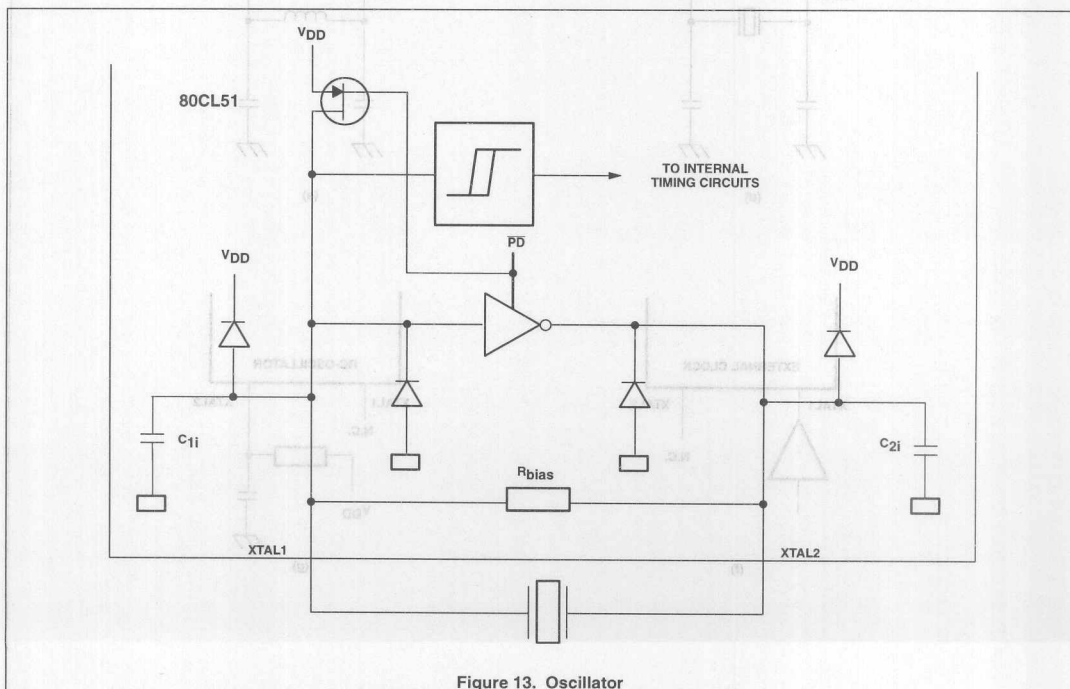


Figure 13. Oscillator

## Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

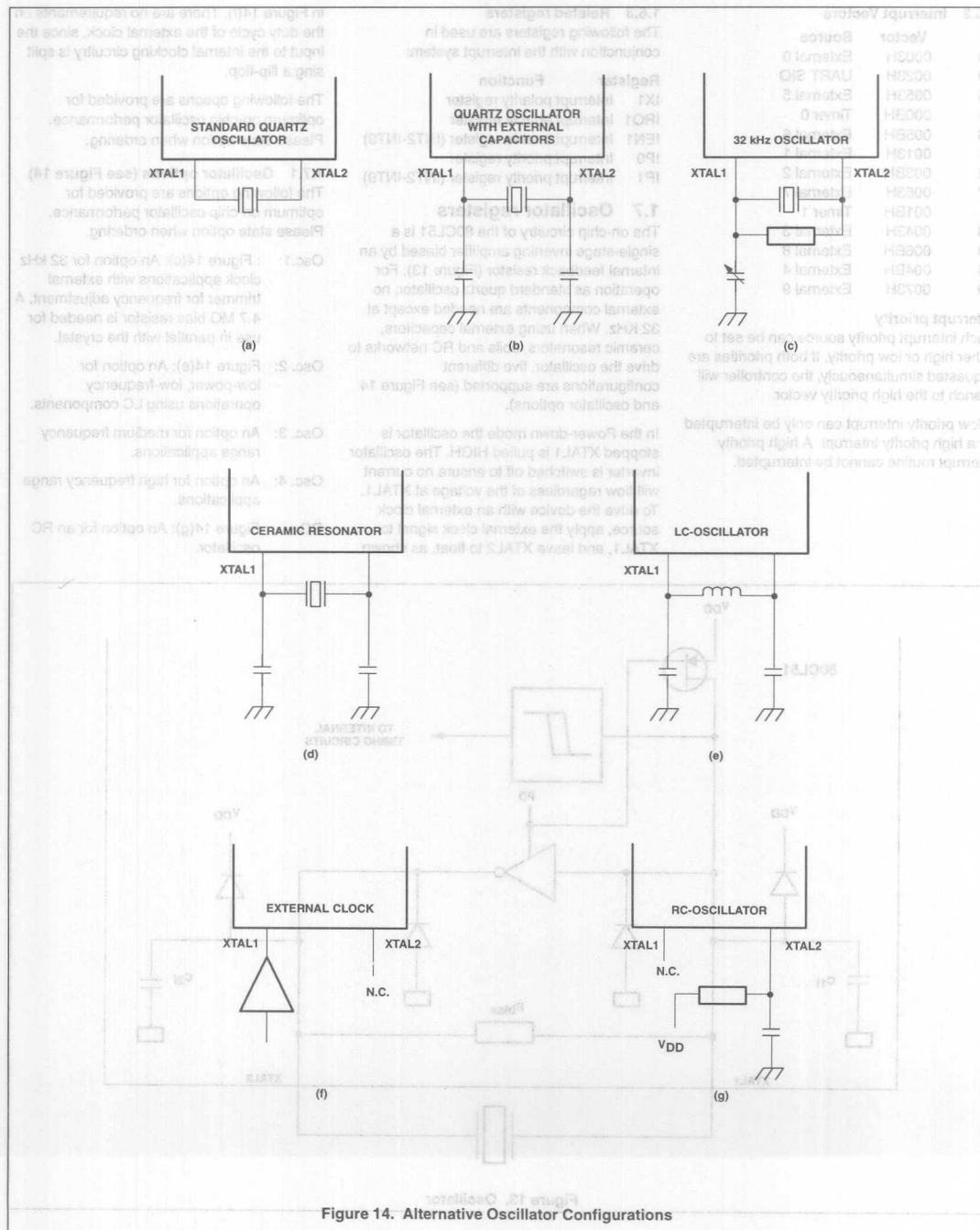


Figure 14. Alternative Oscillator Configurations



# Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

## OSCILLATOR TYPE SELECTION GUIDE

RESONATOR	f(MHz)	OPTION	C1 EXT. (pF)		C2 EXT. (pF)		MAX. RESONATOR SERIES RESISTANCE
			MIN.	MAX.	MIN.	MAX.	
Quartz	0.032	OSC. 1	0	0	5	15	15 k $\Omega$ <sup>1</sup>
Quartz	1.0	OSC. 2	0	30	0	30	600 $\Omega$
Quartz	3.58	OSC. 2	0	15	0	15	100 $\Omega$
Quartz	4.0	OSC. 2	0	20	0	20	75 $\Omega$
Quartz	6.0	OSC. 3	0	10	0	10	60 $\Omega$
Quartz	10.0	OSC. 4	0	15	0	15	60 $\Omega$
Quartz	12.0	OSC. 4	0	10	0	10	40 $\Omega$
Quartz	16.0	OSC. 4	0	15	0	15	20 $\Omega$
PXE	0.455	OSC. 2	40	50	40	50	10 $\Omega$
PXE	1.0	OSC. 2	15	50	15	50	100 $\Omega$
PXE	3.58	OSC. 2	0	40	0	40	10 $\Omega$
PXE	4.0	OSC. 2	0	40	0	40	10 $\Omega$
PXE	6.0	OSC. 2	0	20	0	20	5 $\Omega$
PXE	10.0	OSC. 3	0	15	0	15	6 $\Omega$
PXE	12.0	OSC. 4	10	40	10	40	6 $\Omega$
LC		OSC. 2	20	90	20	90	10 $\mu$ H = 1 $\Omega$ 100 $\mu$ H = 5 $\Omega$ 1 mH = 75 $\Omega$

### NOTES:

- 32 kHz quartz crystals with a series resistance higher than 15 k $\Omega$  will reduce the guaranteed supply voltage range to 2.5 -3.5V.
- The equivalent circuit data of the internal oscillator compares with that of matched crystals.

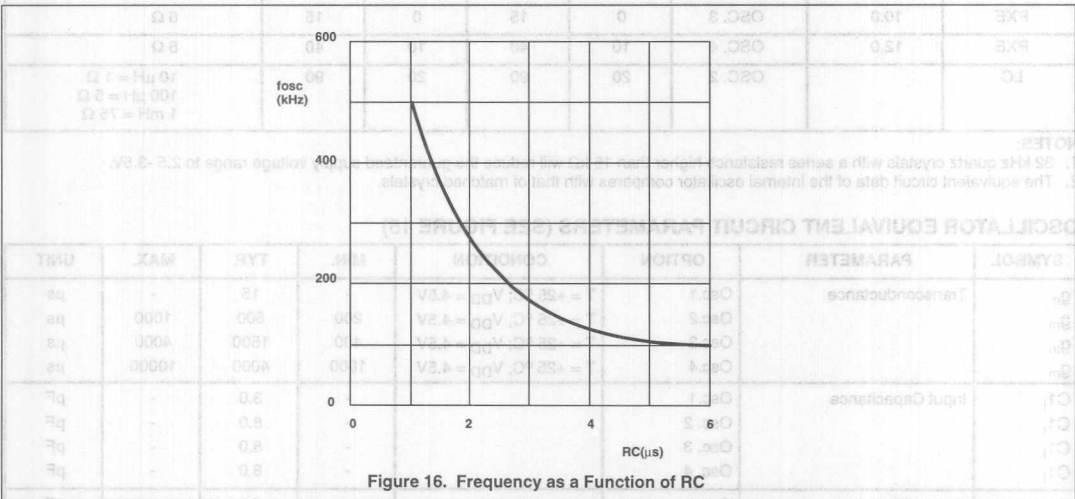
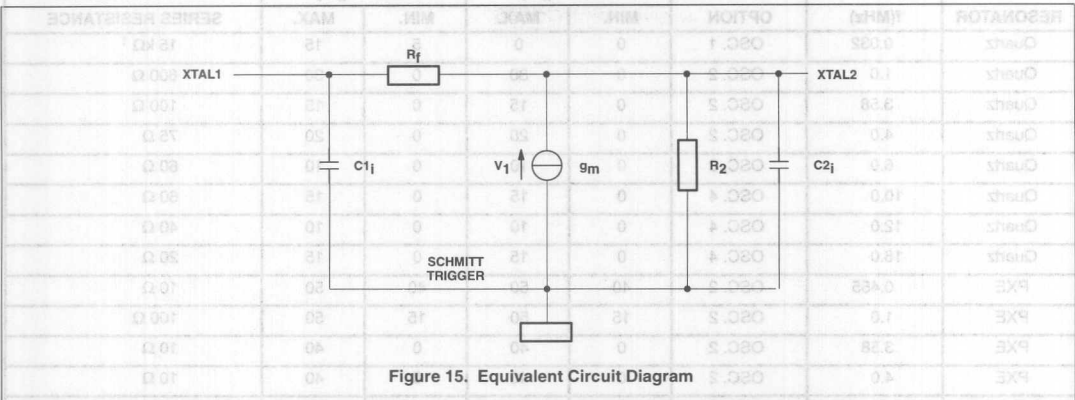
## OSCILLATOR EQUIVALENT CIRCUIT PARAMETERS (SEE FIGURE 15)

SYMBOL	PARAMETER	OPTION	CONDITION	MIN.	TYP.	MAX.	UNIT
$g_m$	Transconductance	Osc.1	T = +25 °C; V <sub>DD</sub> = 4.5V	-	15	-	$\mu$ S
$g_m$		Osc.2	T = +25 °C; V <sub>DD</sub> = 4.5V	200	600	1000	$\mu$ S
$g_m$		Osc.3	T = +25 °C; V <sub>DD</sub> = 4.5V	400	1500	4000	$\mu$ S
$g_m$		Osc.4	T = +25 °C; V <sub>DD</sub> = 4.5V	1000	4000	10000	$\mu$ S
C1 <sub>i</sub>	Input Capacitance	Osc.1		-	3.0	-	pF
C1 <sub>i</sub>		Osc. 2		-	8.0	-	pF
C1 <sub>i</sub>		Osc. 3		-	8.0	-	pF
C1 <sub>i</sub>		Osc. 4		-	8.0	-	pF
C2 <sub>i</sub>	Output Capacitance	Osc.1		-	23	-	pF
C2 <sub>i</sub>		Osc. 2		-	8.0	-	pF
C2 <sub>i</sub>		Osc. 3		-	8.0	-	pF
C2 <sub>i</sub>		Osc. 4		-	8.0	-	pF
R2	Output Capacitance	Osc.1		-	3800	-	k $\Omega$
R2		Osc. 2		-	65	-	k $\Omega$
R2		Osc. 3		-	18	-	k $\Omega$
R2		Osc. 4		-	5.0	-	k $\Omega$



1.7.2 RC Oscillator (see Figure 16)

The externally adjustable RC-oscillator has a frequency range from 100 kHz to 500 kHz.



## Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

### 1.8 Reset Circuitry

To initialize the 80CL51, a reset is performed by either of two methods:

- via the RST pin
- via a power-on-reset

It leaves the internal registers as follows:

REGISTER	CONTENT
ACC	0000 0000
B	0000 0000
DPL	0000 0000
DPH	0000 0000
IEN0	0000 0000
IEN1	0000 0000
IP0	XX00 0000
IP1	0000 0000
IX1	0000 0000
IRQ1	0000 0000
PCH	0000 0000
PCL	0000 0000
PCON	0XXX 0000
PSW	0000 0000
P0-P3	1111 1111
S0BUF	XXXX XXXX
S0CPN	0000 0000
SP	0000 0111
TCON	0000 0000
TH0, TH1	0000 0000
TL0, TH1	0000 0000
TL0, TL1	0000 0000
TMOD	0000 0000

The reset state of the port pins is mask-programmable and can therefore be defined by the user.

The standard reset value for port P0-P3 is 1111 1111.

The reset input to the 80CL51 is RST pin 9. A Schmitt trigger qualifies the input for noise rejection. The output of the Schmitt trigger is

sampled by the reset circuitry every machine cycle.

A reset is accomplished by holding the RST pin HIGH for at least two machine cycles (24 oscillator periods), while the oscillator is running. The CPU responds by generating an internal reset. Port pins adopt their reset state immediately after RST goes HIGH. During reset ALE and PSEN are held HIGH.

The external reset is asynchronous to the internal clock. The RST pin is sampled during State 5, Phase 2 of every machine cycle. After a HIGH is detected at the RST pin, an internal reset is repeated every cycle until RST goes LOW.

The internal RAM is not affected by reset. When  $V_{DD}$  is turned on the RAM contents are indeterminate.

#### 1.8.1 Power-on reset

The 80CL51 contains on-chip circuitry which switch the port pins to the customer defined logic level as soon as  $V_{DD}$  exceeds 1.3V. As soon as the minimum supply voltage is reached, the oscillator will start up. However, to ensure that the oscillator is stable before the controller starts, the clock signals are gated away from the CPU for a further 1536 oscillator periods. During that time the CPU is held in a reset state.

A hysteresis of approximately 50 mV at a typical power-on switching level of 1.3 V will ensure correct operation.

The on-chip Power-on circuitry can be switched off via the mask option "OFF". This option reduces the power-down current to typically 800  $\mu$ A and can be chosen if external reset circuitry is used. For applications not requiring the internal reset option, "OFF" should be chosen.

An automatic reset can be obtained at power-on by connecting the RST pin to  $V_{DD}$  via a 10  $\mu$ F capacitor. At power-on, the voltage on the RST pin is equal to  $V_{DD}$  minus the capacitor voltage, and decreases from  $V_{DD}$  as the capacitor discharges through the internal resistor  $R_{RST}$  to ground. The larger the capacitor, the more slowly  $V_{RST}$  decreases.  $V_{RST}$  must remain above the lower threshold of the Schmitt trigger long enough to effect a complete reset. The time required is the oscillator start-up time, plus 2 machine cycles.

### 1.9 P80CL31: ROMless version of P80CL51

The P80CL31 is a low voltage ROMless version of the P80CL51 microcontroller. The mask options on the P80CL31 are fixed as follows:

- Port options: all ports have option "1S", i.e., standard port, high after reset
- Oscillator option: OSC3
- Power-on Reset option: OFF

### 1.10 P80C51: 5V standard version

The P80C51 is a 5V version of the low voltage P80CL51 microcontroller. All functional features of the P80CL51 are maintained in the P80C51 with the exception of the mask options. The mask options on the P80C51 are as follows:

- Port options: all ports have option "1S", i.e., standard port, high after reset.
- Oscillator options: OSC3
- Power-on Reset option: OFF

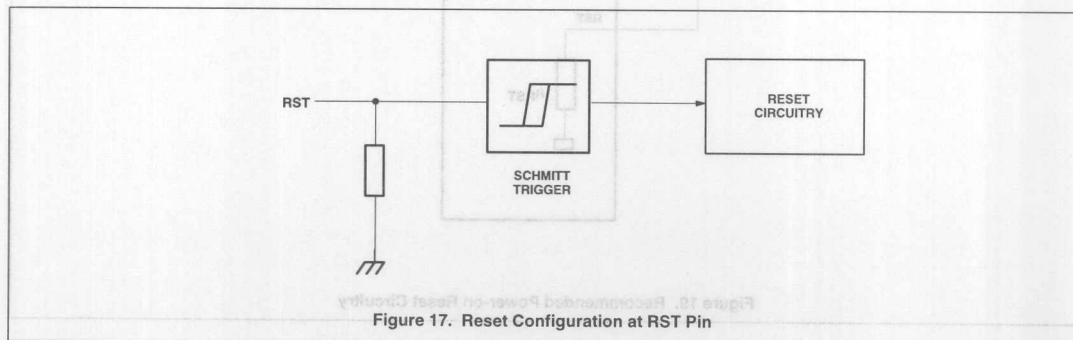


Figure 17. Reset Configuration at RST Pin

Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

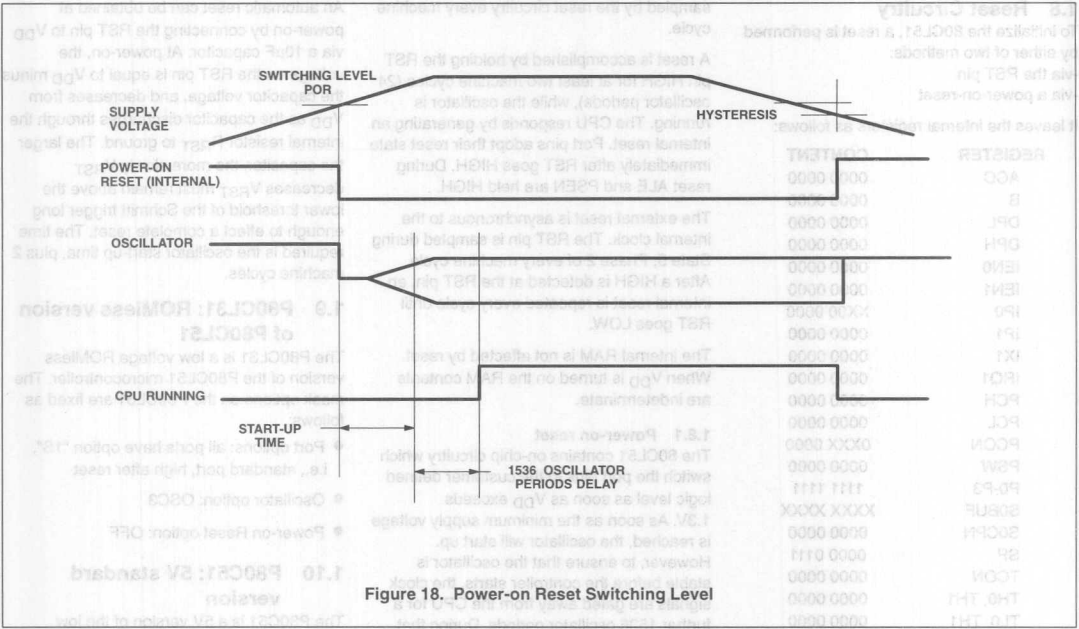


Figure 18. Power-on Reset Switching Level

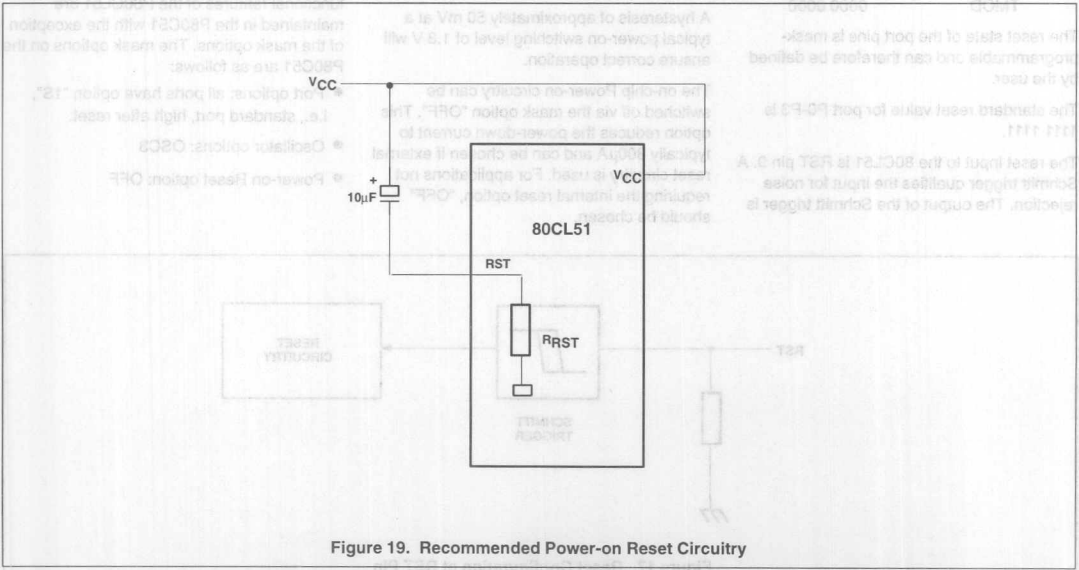


Figure 19. Recommended Power-on Reset Circuitry

### Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

## 2.0 RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

SYMBOL	PARAMETER	MIN.	MAX.	UNIT
V <sub>DD</sub>	Supply voltage (pin 40)	-0.5	+ 6.5	V
V <sub>I</sub>	All input voltages	-0.5	V <sub>DD</sub> +0.5	V
I <sub>I</sub> , I <sub>O</sub>	DC current into any input or output	-	5	mA
P <sub>TOT</sub>	Total power dissipation	-	300	mW
T <sub>STG</sub>	Storage temperature range	-65	+150	°C
T <sub>AMB</sub>	Operating ambient temperature range	-40	+85	°C
T <sub>J</sub>	Operating junction temperature	-	125	°C

### 3.0 DC CHARACTERISTICS P80CL31/P80CL51

$V_{SS} = 0V$ ;  $T_{AMB} = -40$  to  $+85^{\circ}C$ ; all voltages with respect to  $V_{SS}$  unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
$V_{DD}$	Supply voltage	$V_{SS} = 0V$	1.8		6.0	V
$V_{DD}$	RAM retention in power down mode		1.0		—	V
<b>Supply current operating (Note 1, Note 4)</b>						
$I_{DD}$	OSC 1 option	$f_{clk} = 32 \text{ KHz}$ ; $V_{DD} = 1.8V$ $T_{AMB} = 25^{\circ}C$			50	$\mu A$
$I_{DD}$	OSC 2 option	$f_{clk} = 3.58 \text{ MHz}$ ; $V_{DD} = 3V$	—	—	2.5	mA
$I_{DD}$	OSC 3 option	$f_{clk} = 16 \text{ MHz}$ ; $V_{DD} = 5V$	—	—	24	mA
$I_{DD}$	OSC 4 option	$f_{clk} = 16 \text{ MHz}$ ; $V_{DD} = 5V$	—	—	26	mA
<b>Idle Mode (Note 2, Note 4)</b>						
$I_{DD}$	OSC 1 option	$f_{clk} = 32 \text{ KHz}$ ; $V_{DD} = 1.8V$ $T_{AMB} = 25^{\circ}C$	—	—	25	$\mu A$
$I_{DD}$	OSC 2 option	$f_{clk} = 3.58 \text{ MHz}$ ; $V_{DD} = 3V$	—	—	1.0	mA
$I_{DD}$	OSC 3 option	$f_{clk} = 16 \text{ MHz}$ ; $V_{DD} = 5V$	—	—	10	mA
$I_{DD}$	OSC 4 option	$f_{clk} = 16 \text{ MHz}$ ; $V_{DD} = 5V$	—	—	12	mA
$I_{PD}$	Power down (Note 3, Note 4)	$V_{DD} = 1.8V$ , $T_{AMB} = 25^{\circ}C$	—	—	10	$\mu A$
<b>Inputs</b>						
$V_{IL}$	Input voltage LOW		$V_{SS}$	—	$0.3V_{DD}$	V
$V_{IH}$	Input voltage HIGH		$0.7V_{DD}$	—	$V_{DD}$	V
$I_{IL}$	Input current logic 0 (Port 1, 2, 3)	$V_{DD} = 5V$ , $V_{IN} = 0.4V$	—	—	100	$\mu A$
		$V_{DD} = 2.5V$ , $V_{IN} = 0.4V$	—	—	50	$\mu A$
$I_{TL}$	Input current logic 1 to 0 transition (Port 1, 2, 3)	$V_{DD} = 5V$ , $V_{IN} = V_{DD}/2$	—	—	1.0	mA
		$V_{DD} = 2.5V$ , $V_{IN} = V_{DD}/2$	—	—	500	$\mu A$
$+I_{IL}$	Input leakage current (Port 0, EA)	$V_{SS} < V_I < V_{DD}$	—	—	10	$\mu A$
<b>Outputs</b>						
$I_{OL}$	Output sink current LOW	$V_{DD} = 5V$ , $V_{OL} = 0.4V$	1.6	—	—	mA
		$V_{DD} = 2.5V$ , $V_{OL} = 0.4V$	0.7	—	—	mA
$-I_{OH}$	Output source current HIGH (push-pull options only)	$V_{DD} = 5V$ ; $V_{OH} = V_{DD} - 0.4V$	1.6	—	—	mA
		$V_{DD} = 2.5V$ ; $V_{OH} = V_{DD} - 0.4V$	0.7	—	—	mA
$R_{RST}$	RST pull-down resistor		10	—	200	k $\Omega$

#### NOTES:

- The operating supply current is measured with all output pins disconnected; XTAL 1 driven with  $t_r = t_f = 10ns$ ;  $V_{IL} = V_{SS}$ ;  $V_{IH} = V_{DD}$ ; XTAL 2 not connected; EA = RST = Port 0 =  $V_{DD}$ ; all open drain outputs connected to  $V_{SS}$ .
- The idle mode supply current is measured with all output pins disconnected; XTAL 1 driven with  $t_r = t_f = 10ns$ ;  $V_{IL} = V_{SS}$ ; XTAL 2 not connected; EA = Port 0 =  $V_{DD}$ ; RST =  $V_{SS}$ ; all open drain outputs connected to  $V_{SS}$ .
- The power-down current is measured with all output pins disconnected; XTAL 1 not connected; EA = Port 0 =  $V_{DD}$ ; RST =  $V_{SS}$ ; all open drain outputs connected to  $V_{SS}$ .
- Circuits with Power-on Reset option "OFF" are tested at  $V_{DD}$  minimum = 1.8V; with option "ON" (typically 1.3V) they are tested at  $V_{DD}$  minimum = 2.3V. Please note, option "ON" is only available on P80CL51.

Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

4.0 DC CHARACTERISTICS P80C51

V<sub>SS</sub> = 0V; V<sub>DD</sub> = 5V ± 10%; f<sub>clk</sub> = 3.5 to 16MHz; T<sub>AMB</sub> = -40 to +85°C; all voltages with respect to V<sub>SS</sub> unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
V <sub>DD</sub>	Supply voltage	V <sub>SS</sub> = 0V	4.5	-	5.5	V
Supply Current						
I <sub>DD</sub>	Operating (Note 1)	f <sub>CLK</sub> = 16MHz, V <sub>DD</sub> = 5V	-	-	24	mA
I <sub>DD</sub>	Idle mode (Note 2)	f <sub>CLK</sub> = 16MHz, V <sub>DD</sub> = 5V	-	-	10	mA
I <sub>PD</sub>	Power down (Note 3)	V <sub>DD</sub> = 5V	-	-	50	µA
Inputs						
V <sub>IL</sub>	Input voltage LOW		V <sub>SS</sub>	-	0.3V <sub>DD</sub>	V
V <sub>IH</sub>	Input voltage HIGH		0.7V <sub>DD</sub>	-	V <sub>DD</sub>	V
I <sub>IL</sub>	Input current logic 0 (Port 1, 2, 3)	V <sub>IN</sub> = 0.4V	-	-	100	µA
I <sub>IL</sub>	Input current logic 1 to 0 transition (Port 1, 2, 3)	V <sub>IN</sub> = V <sub>DD</sub> /2	-	-	1.0	mA
I <sub>IL</sub>	Input leakage current (Port 0, EA)	V <sub>SS</sub> < V <sub>I</sub> < V <sub>DD</sub>	-	-	10	µA
Outputs						
I <sub>OL</sub>	Output sink current LOW	V <sub>OL</sub> = 0.4V	1.6	-	-	mA
I <sub>OH</sub>	Output source current HIGH (push-pull options only)	V <sub>OH</sub> = V <sub>DD</sub> - 0.4V	1.6	-	-	mA
R <sub>RST</sub>	RST pull-down resistor		10	-	200	kΩ

NOTES:

- The operating supply current is measured with all output pins disconnected; XTAL 1 driven with t<sub>R</sub> = t<sub>F</sub> = 10ns; V<sub>IL</sub> = V<sub>SS</sub>; V<sub>IH</sub> = V<sub>DD</sub>; XTAL 2 not connected; EA = RST = Port 0 = V<sub>DD</sub>; all open drain outputs connected to V<sub>SS</sub>.
- The idle mode supply current is measured with all output pins disconnected; XTAL 1 driven with t<sub>R</sub> = t<sub>F</sub> = 10ns; V<sub>IL</sub> = V<sub>SS</sub>; XTAL 2 not connected; EA = Port 0 = V<sub>DD</sub>; RST = V<sub>SS</sub>; all open drain outputs connected to V<sub>SS</sub>.
- The power-down current is measured with all output pins disconnected; XTAL 1 not connected; EA = Port 0 = V<sub>DD</sub>; RST = V<sub>SS</sub>; all open drain outputs connected to V<sub>SS</sub>.
- Please note, option "ON" is only available on P80CL51.

ns					Address hold time after RD (note 1)
ns					Data hold time after WR
ns					Data setup time before WR
ns					Data valid to WR transition
ns					Time from RD or WR HIGH to ALE HIGH
ns					Time from address to RD and WR
ns					Time from ALE to RD and WR
ns					Address to valid data input
ns					Time from ALE to valid data input
ns					Data float delay after RD
ns					RD to valid data input
ns					Address hold time after ALE

NOTE: 1. Including the 80CL31 or P80C51 to devices with fast times up to 1µs is permitted. This limited bus connection will not cause damage to Port 0 drivers.



# Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

## 5.0 AC CHARACTERISTICS

$V_{DD} = 5\text{ V}$ ;  $V_{SS} = 0\text{ V}$ ;  $T_{amb} = -40\text{ to }+85^\circ\text{C}$ ;  $C_L = 50\text{ pF}$  for Port 0, ALE and PSEN;  $C_L = 40\text{ pF}$  for all other outputs, unless otherwise specified.

### PROGRAM MEMORY (See Figure 20)

SYMBOL	PARAMETER	VARIABLE CLOCK			
		MIN.	TYP.	MAX.	UNIT
$t_{LL}$	ALE pulse duration	$2T_{CK}-40$	-	-	ns
$t_{AL}$	Address set-up time to ALE	$T_{CK}-40$	-	-	ns
$t_{LA}$	Address hold time to ALE	$T_{CK}-35$	-	-	ns
$t_{LC}$	Time from ALE to control pulse PSEN	$T_{CK}-25$	-	-	ns
$t_{LIV}$	Time from ALE to valid instruction input	-	-	$4T_{CK}-100$	ns
$t_{CC}$	Control pulse duration PSEN	$3T_{CK}-35$	-	-	ns
$t_{CIV}$	Time from PSEN to valid instruction input	-	-	$3T_{CK}-125$	ns
$t_{CI}$	Input instruction hold time after PSEN	0	-	-	ns
$t_{CIF}$	Input instruction float delay after PSEN	-	-	$T_{CK}-20$	ns
$t_{AIV}$	Address to valid instruction input	-	-	$5T_{CK}-115$	ns
$t_{AFC}$	Address float time to PSEN	0	-	-	ns

### EXTERNAL DATA MEMORY (See Figures 21 and 22)

SYMBOL	PARAMETER	VARIABLE CLOCK			
		MIN.	TYP.	MAX.	UNIT
$t_{RR}$	RD pulse duration	$6T_{CK}-100$	-	-	ns
$t_{WW}$	WR pulse duration	$6T_{CK}-100$	-	-	ns
$t_{LA}$	Address hold time after ALE	$T_{CK}-35$	-	-	ns
$t_{RD}$	RD to valid data input	$T_{CK}-35$	-	$5T_{CK}-165$	ns
$t_{DFR}$	Data float delay after RD	-	-	$2T_{CK}-70$	ns
$t_{LD}$	Time from ALE to valid data input	-	-	$8T_{CK}-150$	ns
$t_{AD}$	Address to valid data input	-	-	$9T_{CK}-165$	ns
$t_{LW}$	Time from ALE to RD and WR	$3T_{CK}-50$	-	$3T_{CK}+50$	ns
$t_{AW}$	Time from address to RD and WR	$4T_{CK}-130$	-	-	ns
$t_{WHLH}$	Time from RD or WR HIGH to ALE HIGH	$T_{CK}-40$	-	$T_{CK}-40$	ns
$t_{DWX}$	Data valid to WR transition	$T_{CK}-60$	-	-	ns
$t_{DW}$	Data set-up time before WR	$T_{CK}-150$	-	-	ns
$t_{WD}$	Data hold time after WR	$T_{CK}-50$	-	-	ns
$t_{WAFR}$	Address float delay after RD (Note 1)	-	-	12	ns

#### NOTE:

- Interfacing the 80CL51 or P80C51 to devices with float times up to 75ns is permitted. This limited bus connection will not cause damage to Port 0 drivers.

## Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

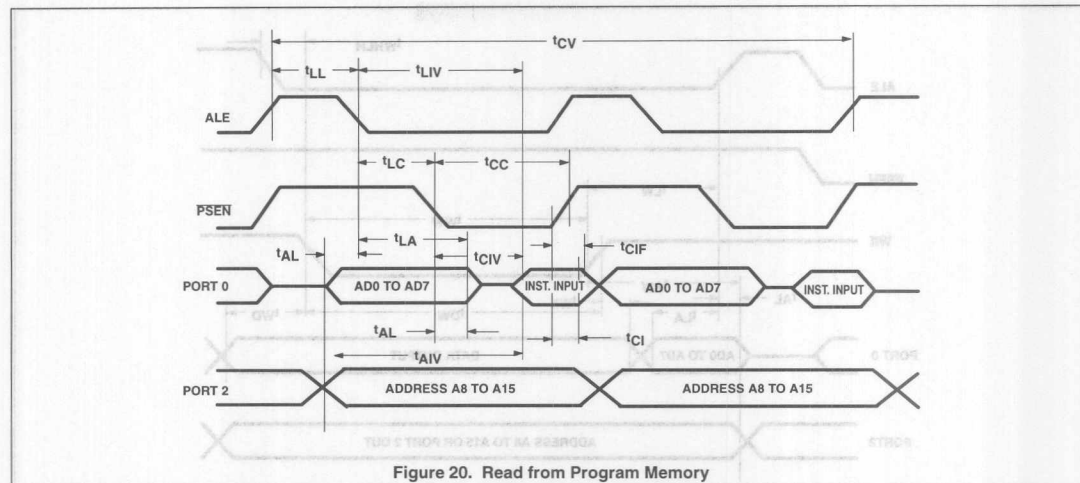


Figure 20. Read from Program Memory

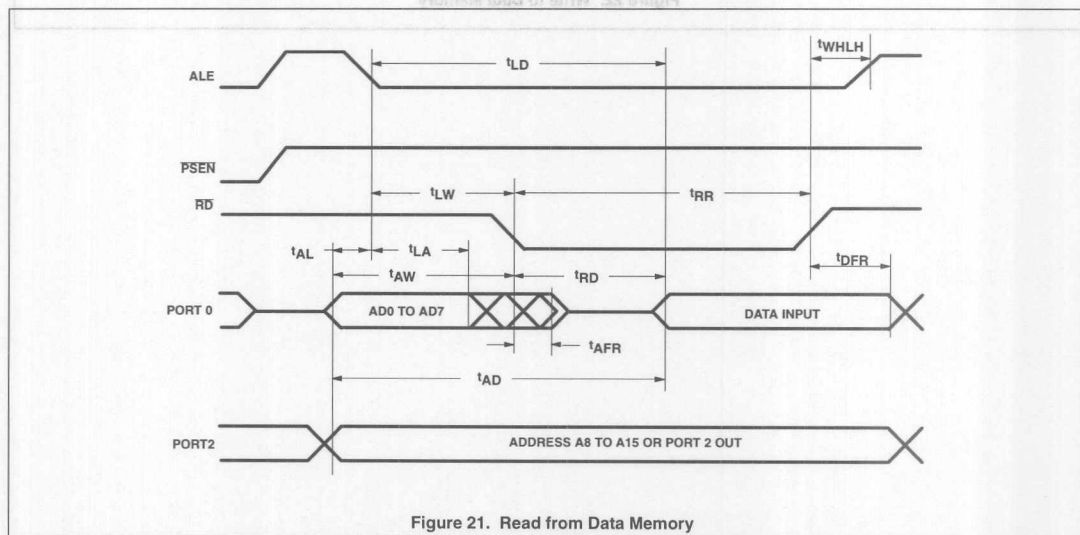


Figure 21. Read from Data Memory

## Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

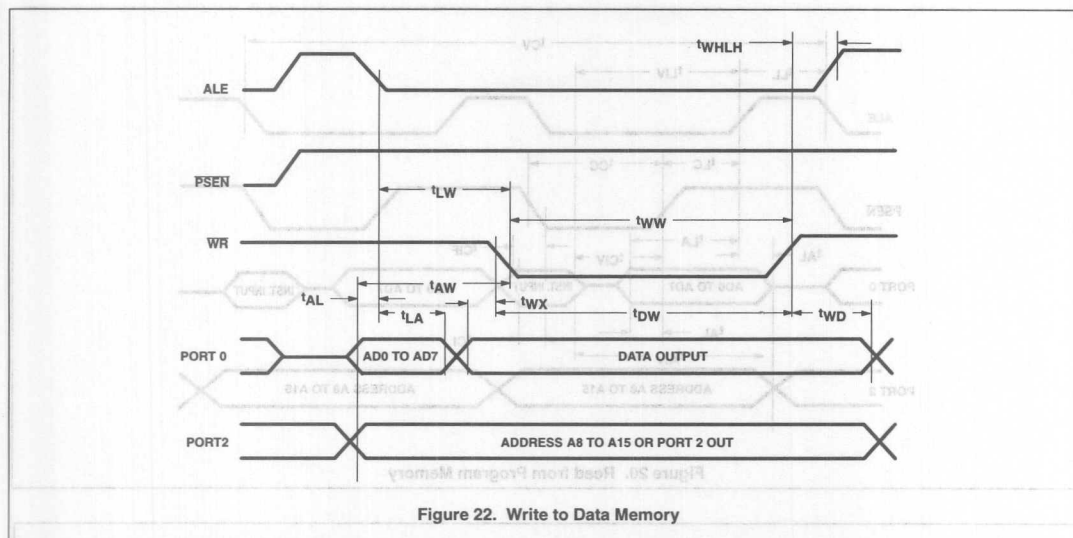


Figure 22. Write to Data Memory

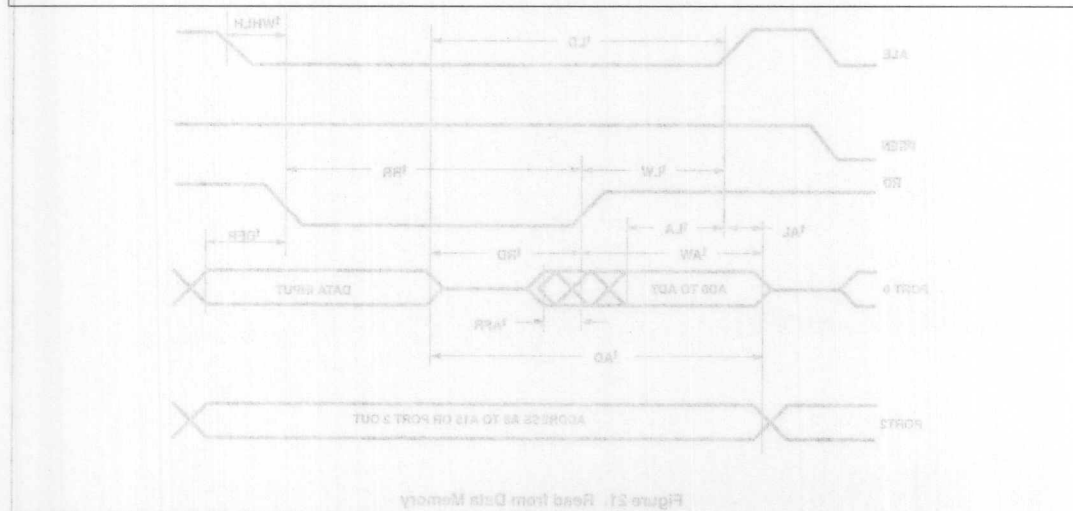


Figure 21. Read from Data Memory

## Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

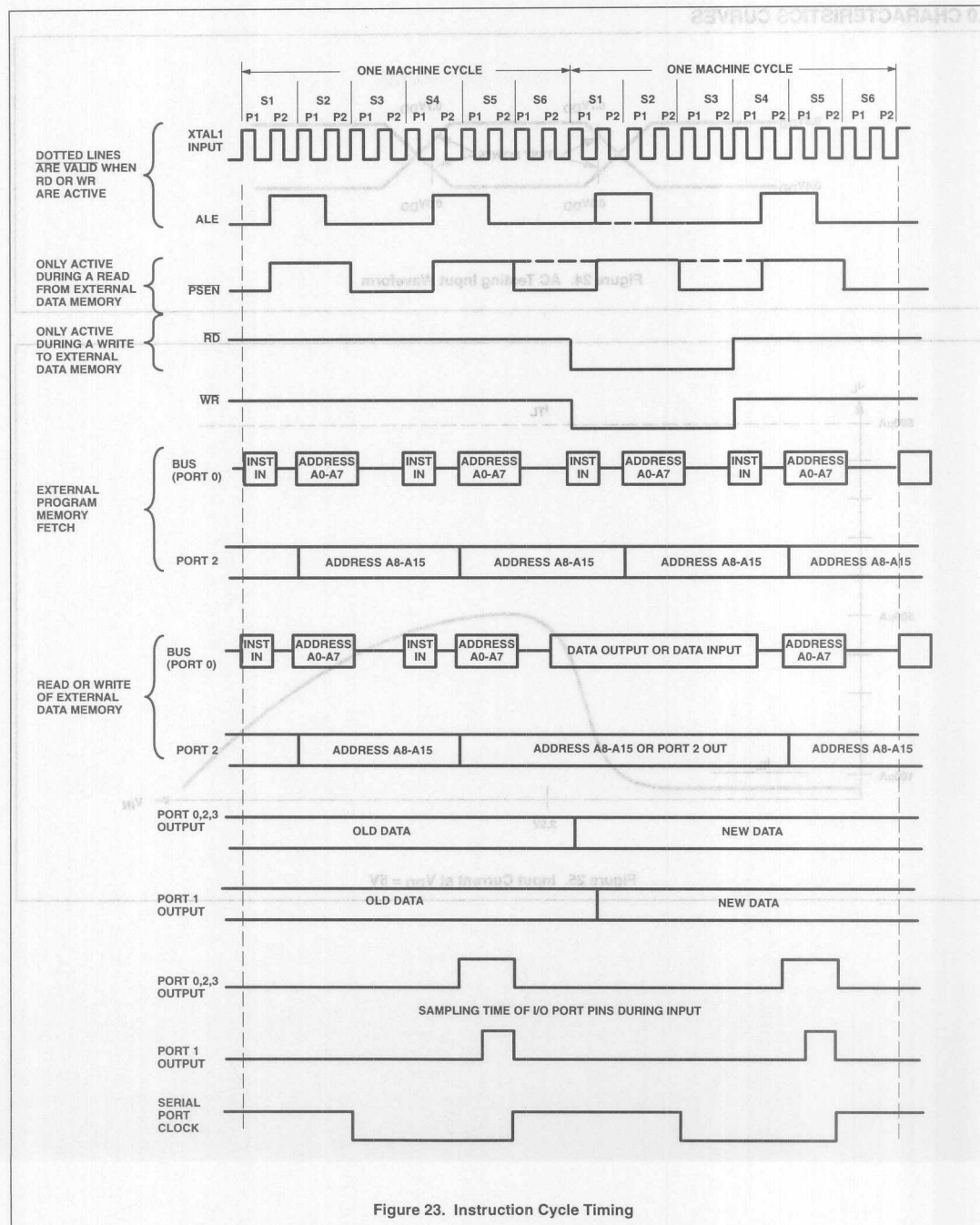
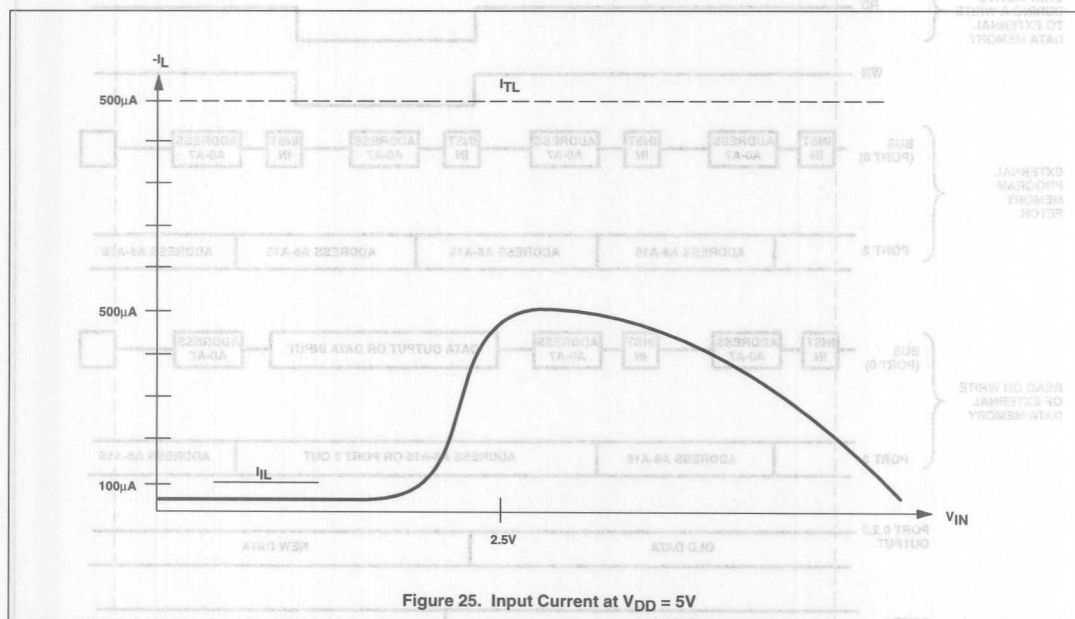
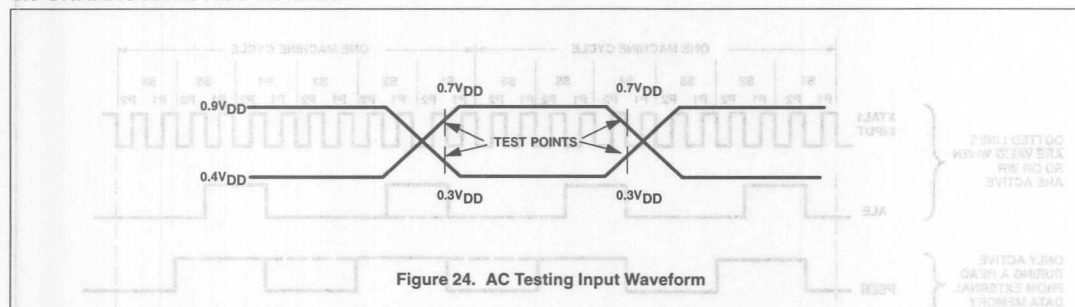


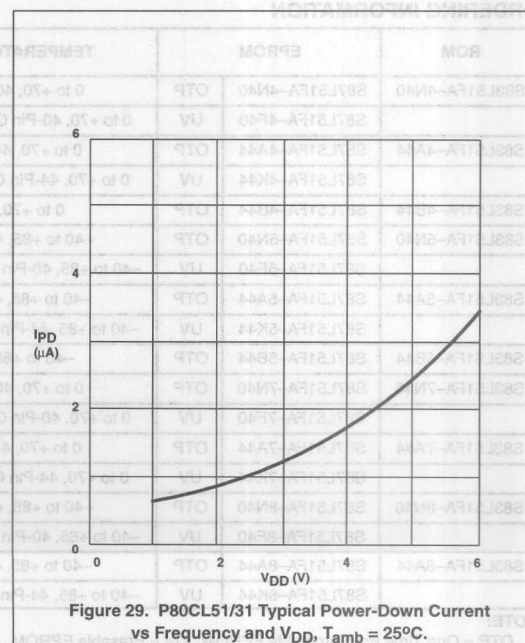
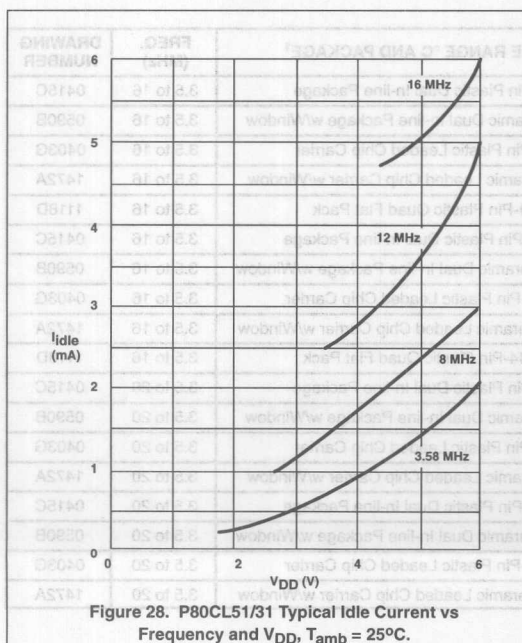
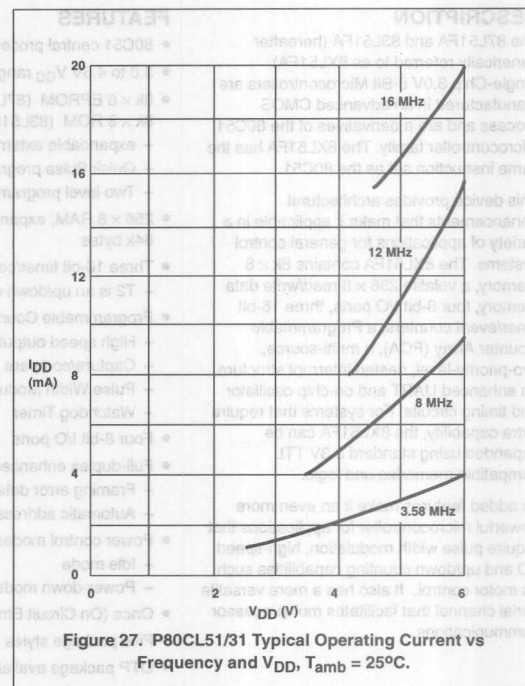
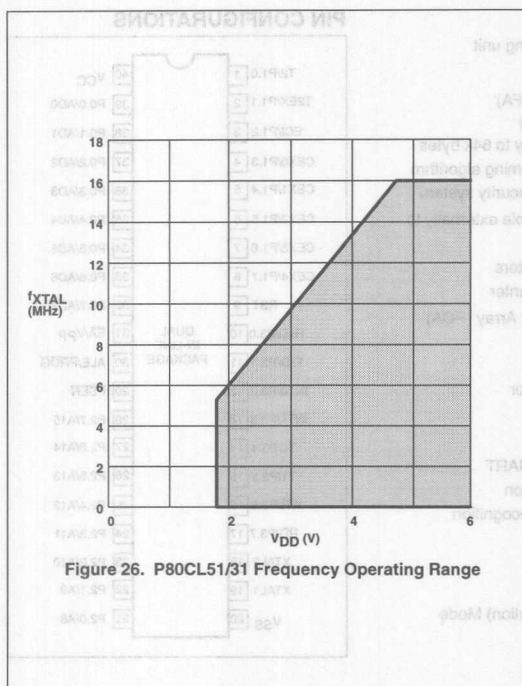
Figure 23. Instruction Cycle Timing

## Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51

## 6.0 CHARACTERISTICS CURVES



## Low-voltage single-chip 8-bit microcontrollers 80CL31/80CL51





## DESCRIPTION

The 87L51FA and 83L51FA (hereafter generically referred to as 8XL51FA) Single-Chip 3.0V 8-Bit Microcontrollers are manufactured in an advanced CMOS process and are a derivatives of the 80C51 microcontroller family. The 8XL51FA has the same instruction set as the 80C51.

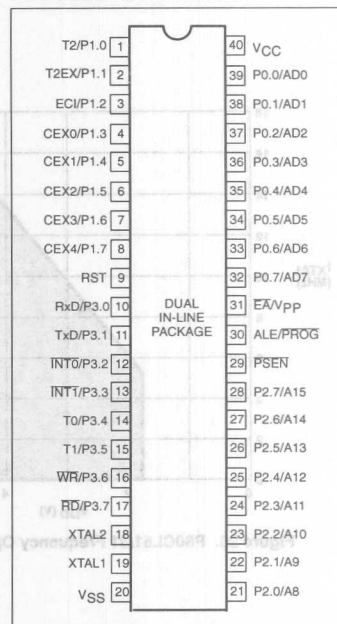
This device provides architectural enhancements that make it applicable in a variety of applications for general control systems. The 8XL51FA contains 8k × 8 memory, a volatile 256 × 8 read/write data memory, four 8-bit I/O ports, three 16-bit timer/event counters, a Programmable Counter Array (PCA), a multi-source, two-priority-level, nested interrupt structure, an enhanced UART and on-chip oscillator and timing circuits. For systems that require extra capability, the 8XL51FA can be expanded using standard 3.3V TTL compatible memories and logic.

Its added features make it an even more powerful microcontroller for applications that require pulse width modulation, high-speed I/O and up/down counting capabilities such as motor control. It also has a more versatile serial channel that facilitates multiprocessor communications.

## FEATURES

- 80C51 central processing unit
- 3.0 to 4.5V  $V_{CC}$  range
- 8k × 8 EPROM (87L51FA)  
8k × 8 ROM (83L51FA)
  - expandable externally to 64k bytes
  - Quick Pulse programming algorithm
  - Two level program security system
- 256 × 8 RAM, expandable externally to 64k bytes
- Three 16-bit timer/counters
  - T2 is an up/down counter
- Programmable Counter Array (PCA)
  - High speed output
  - Capture/compare
  - Pulse Width Modulator
  - Watchdog Timer
- Four 8-bit I/O ports
- Full-duplex enhanced UART
  - Framing error detection
  - Automatic address recognition
- Power control modes
  - Idle mode
  - Power-down mode
- Once (On Circuit Emulation) Mode
- Five package styles
- OTP package available

## PIN CONFIGURATIONS



## ORDERING INFORMATION

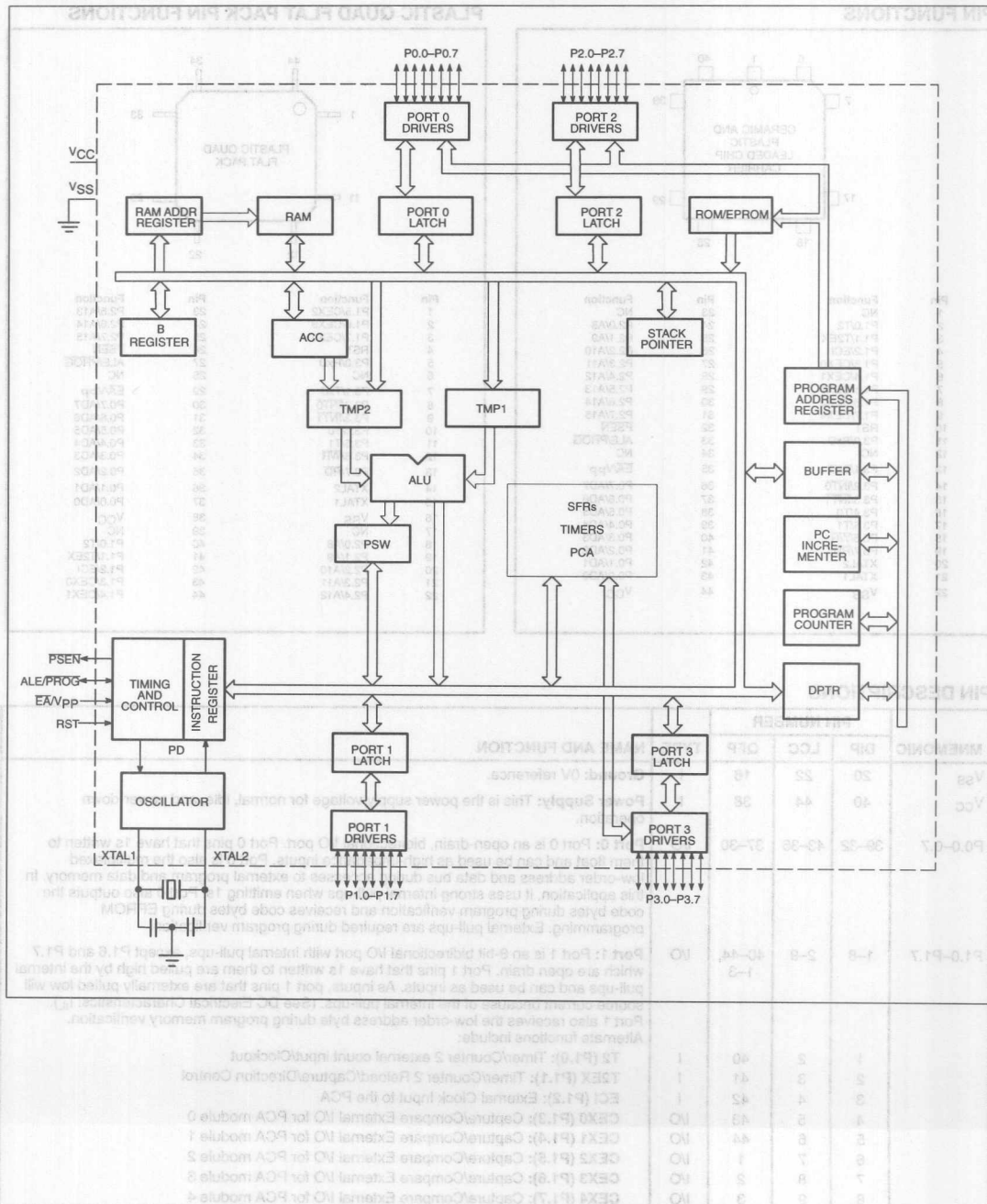
ROM	EPROM		TEMPERATURE RANGE °C AND PACKAGE <sup>1</sup>	FREQ. (MHz)	DRAWING NUMBER
S83L51FA-4N40	S87L51FA-4N40	OTP	0 to +70, 40-Pin Plastic Dual In-line Package	3.5 to 16	0415C
	S87L51FA-4F40	UV	0 to +70, 40-Pin Ceramic Dual In-line Package w/Window	3.5 to 16	0590B
S83L51FA-4A44	S87L51FA-4A44	OTP	0 to +70, 44-Pin Plastic Leaded Chip Carrier	3.5 to 16	0403G
	S87L51FA-4K44	UV	0 to +70, 44-Pin Ceramic Leaded Chip Carrier w/Window	3.5 to 16	1472A
S83L51FA-4B44	S87L51FA-4B44	OTP	0 to +70, 44-Pin Plastic Quad Flat Pack	3.5 to 16	1118D
S83L51FA-5N40	S87L51FA-5N40	OTP	-40 to +85, 40-Pin Plastic Dual In-line Package	3.5 to 16	0415C
	S87L51FA-5F40	UV	-40 to +85, 40-Pin Ceramic Dual In-line Package w/Window	3.5 to 16	0590B
S83L51FA-5A44	S87L51FA-5A44	OTP	-40 to +85, 44-Pin Plastic Leaded Chip Carrier	3.5 to 16	0403G
	S87L51FA-5K44	UV	-40 to +85, 44-Pin Ceramic Leaded Chip Carrier w/Window	3.5 to 16	1472A
S83L51FA-5B44	S87L51FA-5B44	OTP	-40 to +85, 44-Pin Plastic Quad Flat Pack	3.5 to 16	1118D
S83L51FA-7N40	S87L51FA-7N40	OTP	0 to +70, 40-Pin Plastic Dual In-line Package	3.5 to 20	0415C
	S87L51FA-7F40	UV	0 to +70, 40-Pin Ceramic Dual In-line Package w/Window	3.5 to 20	0590B
S83L51FA-7A44	S87L51FA-7A44	OTP	0 to +70, 44-Pin Plastic Leaded Chip Carrier	3.5 to 20	0403G
	S87L51FA-7K44	UV	0 to +70, 44-Pin Ceramic Leaded Chip Carrier w/Window	3.5 to 20	1472A
S83L51FA-8N40	S87L51FA-8N40	OTP	-40 to +85, 40-Pin Plastic Dual In-line Package	3.5 to 20	0415C
	S87L51FA-8F40	UV	-40 to +85, 40-Pin Ceramic Dual In-line Package w/Window	3.5 to 20	0590B
S83L51FA-8A44	S87L51FA-8A44	OTP	-40 to +85, 44-Pin Plastic Leaded Chip Carrier	3.5 to 20	0403G
	S87L51FA-8K44	UV	-40 to +85, 44-Pin Ceramic Leaded Chip Carrier w/Window	3.5 to 20	1472A

### NOTE:

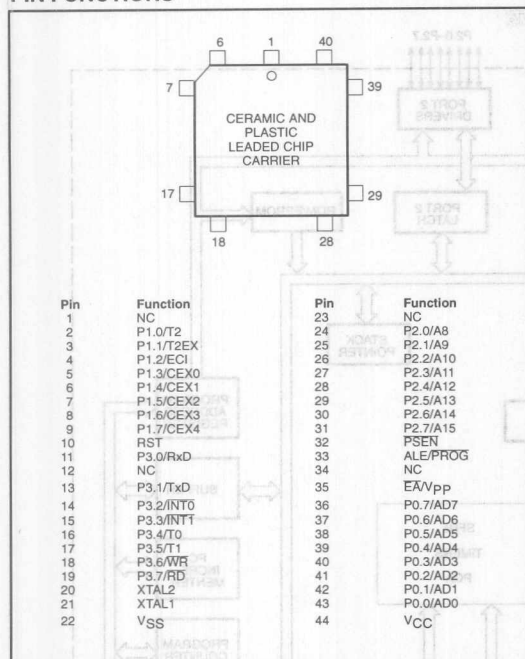
1. OTP = One Time Programmable EPROM. UV = Erasable EPROM.

## CMOS single-chip 3.0V 8-bit microcontroller 83L51FA/87L51FA

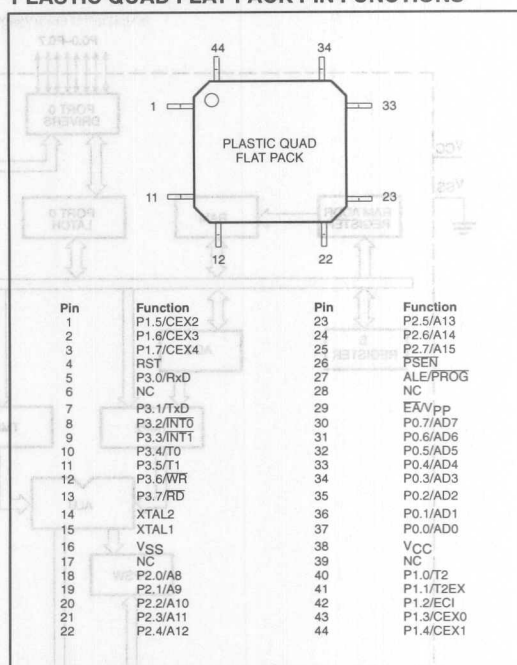
## BLOCK DIAGRAM



## CMOS single-chip 3.0V 8-bit microcontroller 83L51FA/87L51FA

CERAMIC AND PLASTIC LEADED CHIP CARRIER  
PIN FUNCTIONS

## PLASTIC QUAD FLAT PACK PIN FUNCTIONS



## PIN DESCRIPTIONS

MNEMONIC	PIN NUMBER			TYPE	NAME AND FUNCTION
	DIP	LCC	QFP		
V <sub>SS</sub>	20	22	16	I	<b>Ground:</b> 0V reference.
V <sub>CC</sub>	40	44	38	I	<b>Power Supply:</b> This is the power supply voltage for normal, idle, and power-down operation.
P0.0–0.7	39–32	43–36	37–30	I/O	<b>Port 0:</b> Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s. Port 0 also outputs the code bytes during program verification and receives code bytes during EPROM programming. External pull-ups are required during program verification.
P1.0–P1.7	1–8	2–9	40–44, 1–3	I/O	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pull-ups, except P1.6 and P1.7 which are open drain. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 1 also receives the low-order address byte during program memory verification. Alternate functions include: <b>T2 (P1.0):</b> Timer/Counter 2 external count input/Clockout <b>T2EX (P1.1):</b> Timer/Counter 2 Reload/Capture/Direction Control <b>ECI (P1.2):</b> External Clock Input to the PCA <b>CEX0 (P1.3):</b> Capture/Compare External I/O for PCA module 0 <b>CEX1 (P1.4):</b> Capture/Compare External I/O for PCA module 1 <b>CEX2 (P1.5):</b> Capture/Compare External I/O for PCA module 2 <b>CEX3 (P1.6):</b> Capture/Compare External I/O for PCA module 3 <b>CEX4 (P1.7):</b> Capture/Compare External I/O for PCA module 4
	1	2	40	I	
	2	3	41	I	
	3	4	42	I	
	4	5	43	I/O	
	5	6	44	I/O	
	6	7	1	I/O	
	7	8	2	I/O	
	8	9	3	I/O	

## CMOS single-chip 3.0V 8-bit microcontroller 83L51FA/87L51FA

## PIN DESCRIPTIONS (Continued)

MNEMONIC	PIN NUMBER			TYPE	NAME AND FUNCTION
	DIP	LCC	QFP		
P2.0–P2.7	21–28	24–31	18–25	I/O	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: $I_{IL}$ ). Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register. Some Port 2 pins receive the high order address bits during EPROM programming and verification.
P3.0–P3.7	10–17	11–19	5, 7–13	I/O	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: $I_{IL}$ ). Port 3 also serves the special features of the 80C51 family, as listed below: <b>RxD (P3.0):</b> Serial input port <b>TxD (P3.1):</b> Serial output port <b>INT0 (P3.2):</b> External interrupt <b>INT1 (P3.3):</b> External interrupt <b>T0 (P3.4):</b> Timer 0 external input <b>T1 (P3.5):</b> Timer 1 external input <b>WR (P3.6):</b> External data memory write strobe <b>RD (P3.7):</b> External data memory read strobe
RST	9	10	4	I	<b>Reset:</b> A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to $V_{SS}$ permits a power-on reset using only an external capacitor to $V_{CC}$ .
ALE/PROG	30	33	27	I/O	<b>Address Latch Enable/Program Pulse:</b> Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory. This pin is also the program pulse input (PROG) during EPROM programming.
PSEN	29	32	26	O	<b>Program Store Enable:</b> The read strobe to external program memory. When the 83L51FA is executing code from the external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory. PSEN is not activated during fetches from internal program memory.
EA/V <sub>PP</sub>	31	35	29	I	<b>External Access Enable/Programming Supply Voltage:</b> EA must be externally held low to enable the device to fetch code from external program memory locations 0000H and 1FFFFH. If EA is held high, the device executes from internal program memory unless the program counter contains an address greater than 1FFFFH. This pin also receives the 12.75V programming supply voltage ( $V_{PP}$ ) during EPROM programming. If security bit 1 is programmed, EA will be internally latched on Reset.
XTAL1	19	21	15	I	<b>Crystal 1:</b> Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	20	14	O	<b>Crystal 2:</b> Output from the inverting oscillator amplifier.

**NOTE:**

To avoid "latch-up" effect at power-on, the voltage on any pin at any time must not be higher than  $V_{CC} + 0.5V$  or  $V_{SS} - 0.5V$ , respectively.

**TIMER 2**

This is a 16-bit up or down counter, which can be operated as either a timer or event counter. It can be operated in one of three different modes (autoreload, capture or as the baud rate generator for the UART).

In the autoreload mode the Timer can be set to count up or down by setting or clearing the bit DCEN in the T2CON Special Function Register. The SFR's RCAP2H and RCAP2L are used to reload the Timer upon overflow or a 1-to-0 transition on the T2EX input (P1.1).

In the Capture mode Timer 2 can either set TF2 and generate an interrupt or capture its value. To capture Timer 2 in response to a 1-to-0 transition on the T2EX input, the EXEN2 bit in the T2CON must be set. Timer 2 is then captured in SFR's RCAP2H and RCAP2L.

As the baud rate generator, Timer 2 is selected by setting TCLK and/or RCLK in T2CON. As the baud rate generator Timer 2 is incremented at  $1/2$  the oscillator frequency.

**ENHANCED UART**

The 83L51FA UART has all of the capabilities of the standard 80C51 UART plus Framing Error Detection and Automatic Address Recognition. As in the 80C51, all four modes of operation are supported as well as the 9th bit in modes 2 and 3 that can be used to facilitate multiprocessor communication.

The Framing Error Detection allows the UART to look for missing stop bits. If a Stop



## CMOS single-chip 3.0V 8-bit microcontroller 83L51FA/87L51FA

bit is missing, the FE bit in the SCON SFR is set. The FE bit can be checked after each transmission to detect communication errors. The FE bit can only be cleared by software and is not affected by a valid stop bit.

Automatic Address Recognition is used to reduce the CPU service time for the serial port. The CPU only needs to service the UART when it is addressed and, with this done by the on-chip circuitry, the need for software overhead is greatly reduced. This mode works similar to the 9-bit communication mode, except that it uses only 8 bits and the Stop bit is used to cause the RI bit to be set. There are two SFRs associated with this mode. They are SADDR, which holds the slave address and SADEN, which contains a mask that allows selective masking of the slave address so that broadcast addresses can be used.

### PROGRAMMABLE COUNTER ARRAY

The PCA is a sophisticated free-running 16 bit Timer/Counter that drives 5 modules that can be individually configured as Capture inputs, software timers, high speed outputs, or pulse width modulated outputs. In addition, module 4 can be configured as a software controlled watchdog timer.

The Timer portion of the PCA can be configured to run in one of four different modes. The modes are:  $1/2$  the oscillator frequency,  $1/4$  the oscillator frequency, Timer 0 overflows, or from the ECI input.

For the Capture/Compare mode each of the modules has a pair of registers associated with it called CCAPnH and CCAPnL (where  $n = 0, 1, 2, 3, 4$  depending on the module). Both positive and negative transitions can be captured. This means that the PCA has the flexibility to measure phase differences, duty cycles, pulse widths and a wide variety of other digital pulse characteristics.

In the 16-bit software timer mode each of the modules can generate an interrupt upon a compare.

For applications that require accurate pulse widths and edges the PCA modules can be used as High Speed Outputs (HSO). The PCA toggles the appropriate CEXn pin when there is a match between the PCA timer and the modules compare registers.

The pulse width modulator mode for the PCA allows the conversion of digital information into analog signals. Each of the 5 modules can be used in this mode. The frequency of the PWM depends on the clock source for the PCA. The 8-bit PWM output is generated

by comparing the low byte of the PCA (CL) with the module's CCAPnL SFR. When  $CL < CCAPnL$ , the output is high. When  $CL > CCAPnL$ , the output is low.

### POWER OFF FLAG

The Power Off Flag (POF) is set by on-chip circuitry when the  $V_{CC}$  level on the 83L51FA rises from 0 to 3.3V. The POF bit can be set or cleared by software allowing a user to determine if the reset is the result of a power-on or a warm start after powerdown. The  $V_{CC}$  level must remain above 2.0V for the POF to remain unaffected by the  $V_{CC}$  level.

### OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

### Reset

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. To insure a good power-on reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few milliseconds) plus two machine cycles. At power-on, the voltage on  $V_{CC}$  and RST must come up at the same time for a proper start-up.

### Idle Mode

In the idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

### Power-Down Mode

To save even more power, a Power Down mode can be invoked by software. In this mode, the oscillator is stopped and the instruction that invoked Power Down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power Down mode is terminated.

On the 83L51FA either a hardware reset or external interrupt can use an exit from Power Down. Reset redefines all the SFRs but does not change the on-chip RAM. An external interrupt allows both the SFRs and the on-chip RAM to retain their values.

To properly terminate Power Down the reset or external interrupt should not be executed before  $V_{CC}$  is restored to its normal operating level and must be held active long enough for the oscillator to restart and stabilize (normally less than 10ms).

With an external interrupt, INT0 and INT1 must be enabled and configured as level-sensitive. Holding the pin low restarts the oscillator but bringing the pin back high completes the exit. Once the interrupt is serviced, the next instruction to be executed after RETI will be the one following the instruction that put the device into Power Down.

### Design Consideration

- When the idle mode is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

### ONCE™ Mode

The ONCE ("On-Circuit Emulation") Mode facilitates testing and debugging of systems using the 83L51FA without the 83L51FA having to be removed from the circuit. The ONCE Mode is invoked by:

- Pull ALE low while the device is in reset and PSEN is high;
- Hold ALE low as RST is deactivated.

While the device is in ONCE Mode, the Port 0 pins go into a float state, and the other port pins and ALE and PSEN are weakly pulled high. The oscillator circuit remains active. While the 83L51FA is in this mode, an emulator or test CPU can be used to drive the circuit. Normal operation is restored when a normal reset is applied.

## CMOS single-chip 3.0V 8-bit microcontroller 83L51FA/87L51FA

Table 1. External Pin Status During Idle and Power-Down Mode

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>

PARAMETER	RATING	UNIT
Operating temperature under bias	0 to +70 or -40 to +85	°C
Storage temperature range	-65 to +150	°C
Voltage on $\overline{EA}/V_{PP}$ pin to $V_{SS}$	0 to +13.0	V
Voltage on any other pin to $V_{SS}$	-0.5 to +6.5	V
Maximum $I_{OL}$ per I/O pin	15	mA
Power dissipation (based on package heat transfer limitations, not device power consumption)	1.5	W

## NOTES:

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
- This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
- Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to  $V_{SS}$  unless otherwise noted.

## Electrical Deviations from Commercial Specifications for Extended Temperature Range

DC and AC parameters not included here are the same as in the commercial temperature range table.



## CMOS single-chip 3.0V 8-bit microcontroller 83L51FA/87L51FA

## DC ELECTRICAL CHARACTERISTICS

T<sub>amb</sub> = 0°C to +70°C, -40 to +85°C, V<sub>CC</sub> = 3.0V to 4.5V, V<sub>SS</sub> = 0V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP <sup>1</sup>	MAX	
V <sub>IL</sub>	Input low voltage, except EA <sup>2,3</sup>		-0.5		0.8	V
V <sub>IL1</sub>	Input low voltage to EA <sup>2,3</sup>		0		0.8	V
V <sub>IH</sub>	Input high voltage, except XTAL1, RST <sup>2,4</sup>		2.0		V <sub>CC</sub> +0.5	V
V <sub>IH1</sub>	Input high voltage, XTAL1, RST <sup>2,4</sup>		0.7V <sub>CC</sub>		V <sub>CC</sub> +0.5	V
V <sub>OL</sub>	Output low voltage, ports 1, 2, 3 <sup>5</sup>	I <sub>OL</sub> = 1.6mA <sup>6</sup>			0.45	V
V <sub>OL1</sub>	Output low voltage, port 0, ALE, PSEN <sup>5</sup>	I <sub>OL</sub> = 3.2mA <sup>6</sup>			0.45	V
V <sub>OH</sub>	Output high voltage, ports 1, 2, 3, ALE, PSEN <sup>7</sup>	I <sub>OH</sub> = -20µA	V <sub>CC</sub> - 0.5			V
V <sub>OH1</sub>	Output high voltage (port 0 in external bus mode), ALE <sup>8</sup> , PSEN <sup>7</sup>	I <sub>OH</sub> = -3.2mA	V <sub>CC</sub> - 0.7			V
I <sub>IL</sub>	Logical 0 input current, ports 1, 2, 3 <sup>2</sup>	V <sub>IN</sub> = 0.45V			-50	µA
I <sub>TL</sub>	Logical 1-to-0 transition current, ports 1, 2, 3 <sup>2</sup>	See note 9			-650	µA
I <sub>LI</sub>	Input leakage current, port 0	0.45 V <sub>IN</sub> < V <sub>CC</sub> - 0.3			±5	µA
I <sub>CC</sub>	Power supply current: <sup>2</sup> Active mode @ 20MHz <sup>10</sup> Idle mode @ 20MHz Power-down mode	See note 11		9 2 10	22 6 75	mA mA µA
R <sub>RST</sub>	Internal reset pull-down resistor		40		225	kΩ
C <sub>IO</sub>	Pin capacitance <sup>12</sup> (except EA)				15	pF

## NOTES:

- Typical ratings are not guaranteed. The values listed are at room temperature, 3.3V.
- These values apply only to T<sub>amb</sub> = 0°C to +70°C.
- For V<sub>CC</sub> voltages above 3.6V and less than 5.5V, V<sub>IL</sub> = 0.3V<sub>CC</sub> - 0.1
- For V<sub>CC</sub> voltages above 3.6V and less than 5.5V, V<sub>IH</sub> = 0.3V<sub>CC</sub> + .92
- Under steady state (non-transient) conditions, I<sub>OL</sub> must be externally limited as follows:  
Maximum I<sub>OL</sub> per port pin: 15mA (\*NOTE: This is 85°C specification.)  
Maximum I<sub>OL</sub> per 8-bit port: 26mA  
Maximum total I<sub>OL</sub> for all outputs: 71mA  
If I<sub>OL</sub> exceeds the test condition, V<sub>OL</sub> may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
- Capacitive loading on ports 0 and 2 may cause spurious noise to be superimposed on the V<sub>OL</sub>s of ALE and ports 1 and 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100pF), the noise pulse on the ALE pin may exceed 0.8V. In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input. I<sub>OL</sub> can exceed these conditions provided that no single output sinks more than 5mA and no more than two outputs exceed the test conditions.
- Capacitive loading on ports 0 and 2 may cause the V<sub>OH</sub> on ALE and PSEN to momentarily fall below the 0.9V<sub>CC</sub> specification when the address bits are stabilizing.
- ALE is tested to V<sub>OH1</sub>, except when ALE is off then V<sub>OH</sub> is the voltage specification.
- Pins of ports 1, 2 and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when V<sub>IN</sub> is approximately 1.5V.
- I<sub>CCMAX</sub> at other frequencies is given by: Active mode: I<sub>CCMAX</sub> = 0.8 × FREQ + 6; Idle mode: I<sub>CCMAX</sub> = 0.19 × FREQ + 2.50, where FREQ is the external oscillator frequency in MHz. I<sub>CCMAX</sub> is given in mA. See Figure 8.
- See Figures 9 through 12 for I<sub>CC</sub> test conditions.
- Pin capacitance is less than 25pF. Pin capacitance of ceramic package is less than 15pF (except EA it is 25pF). These values are guaranteed by design and are not tested.

## CMOS single-chip 3.0V 8-bit microcontroller 83L51FA/87L51FA

## AC ELECTRICAL CHARACTERISTICS

$T_{amb} = 0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$ ,  $-40$  to  $+85^{\circ}\text{C}$ ,  $V_{CC} = 3.0\text{V}$  to  $4.5\text{V}$ ,  $V_{SS} = 0\text{V}^{1, 2, 3}$

SYMBOL	FIGURE	PARAMETER	16MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
$1/t_{CLCL}$	1	Oscillator frequency			3.5	16	MHz
					3.5	20	MHz
$t_{LHLL}$	1	ALE pulse width	85		$2t_{CLCL}$ -40		ns
$t_{AVLL}$	1	Address valid to ALE low	22		$t_{CLCL}$ -40		ns
$t_{LLAX}$	1	Address hold after ALE low	32		$t_{CLCL}$ -30		ns
$t_{LLIV}$	1	ALE low to valid instruction in		150		$4t_{CLCL}$ -100	ns
$t_{LLPL}$	1	ALE low to PSEN low	32		$t_{CLCL}$ -30		ns
$t_{PLPH}$	1	PSEN pulse width	142		$3t_{CLCL}$ -45		ns
$t_{PLIV}$	1	PSEN low to valid instruction in		82		$3t_{CLCL}$ -105	ns
$t_{PXIX}$	1	Input instruction hold after PSEN	0		0		ns
$t_{PXIZ}$	1	Input instruction float after PSEN		37		$t_{CLCL}$ -25	ns
$t_{AVIV}$	1	Address to valid instruction in		207		$5t_{CLCL}$ -105	ns
$t_{PLAZ}$	1	PSEN low to address float		10		10	ns
<b>Data Memory</b>							
$t_{RLRH}$	2, 3	RD pulse width	275		$6t_{CLCL}$ -100		ns
$t_{WLWH}$	2, 3	WR pulse width	275		$6t_{CLCL}$ -100		ns
$t_{RLDV}$	2, 3	RD low to valid data in		147		$5t_{CLCL}$ -165	ns
$t_{RHDX}$	2, 3	Data hold after RD	0		0		ns
$t_{RHDZ}$	2, 3	Data float after RD		65		$2t_{CLCL}$ -60	ns
$t_{LLDV}$	2, 3	ALE low to valid data in		350		$8t_{CLCL}$ -150	ns
$t_{AVDV}$	2, 3	Address to valid data in		397		$9t_{CLCL}$ -165	ns
$t_{LLWL}$	2, 3	ALE low to RD or WR low	137	237	$3t_{CLCL}$ -50	$3t_{CLCL}$ +50	ns
$t_{AVWL}$	2, 3	Address valid to WR low or RD low	175		$4t_{CLCL}$ -75		ns
$t_{QVWX}$	2, 3	Data valid to WR transition	42		$t_{CLCL}$ -20		ns
$t_{WHQX}$	2, 3	Data hold after WR	42		$t_{CLCL}$ -20		ns
$t_{QVWH}$	3	Data valid to WR high	287		$7t_{CLCL}$ -150		ns
$t_{RLAZ}$	2, 3	RD low to address float		0		0	ns
$t_{WHLH}$	2, 3	RD or WR high to ALE high	40	87	$t_{CLCL}$ -20	$t_{CLCL}$ +25	ns
<b>External Clock</b>							
$t_{CHCX}$	5	High time	12		20		ns
$t_{CLCX}$	5	Low time	12		20		ns
$t_{CLCH}$	5	Rise time		20		20	ns
$t_{CHCL}$	5	Fall time		20		20	ns
<b>Shift Register</b>							
$t_{XLXL}$	4	Serial port clock cycle time	1		$12t_{CLCL}$		$\mu\text{s}$
$t_{QVXH}$	4	Output data setup to clock rising edge	492		$10t_{CLCL}$ -133		ns
$t_{XHGX}$	4	Output data hold after clock rising edge	8		$2t_{CLCL}$ -117		ns
$t_{XHDX}$	4	Input data hold after clock rising edge	0		0		ns
$t_{XHDX}$	4	Clock rising edge to input data valid		492		$10t_{CLCL}$ -133	ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- Interfacing the 83L51FA to devices with float times up to 45ns is permitted. This limited bus contention will not cause damage to Port 0 drivers.

## CMOS single-chip 3.0V 8-bit microcontroller 83L51FA/87L51FA

## EXPLANATION OF THE AC SYMBOLS

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

A – Address

C – Clock

D – Input data

H – Logic level high

I – Instruction (program memory contents)

L – Logic level low, or ALE

P – PSEN

Q – Output data

R – RD signal

t – Time

V – Valid

W – WR signal

X – No longer a valid logic level

Z – Float

**Examples:**  $t_{AVLL}$  = Time for address valid to

ALE low.

$t_{LLPL}$  = Time for ALE low to PSEN low.

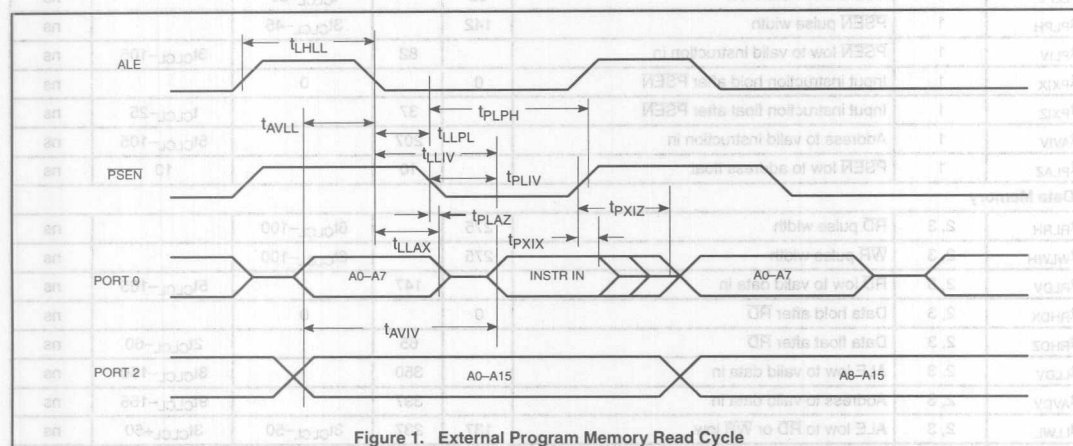


Figure 1. External Program Memory Read Cycle

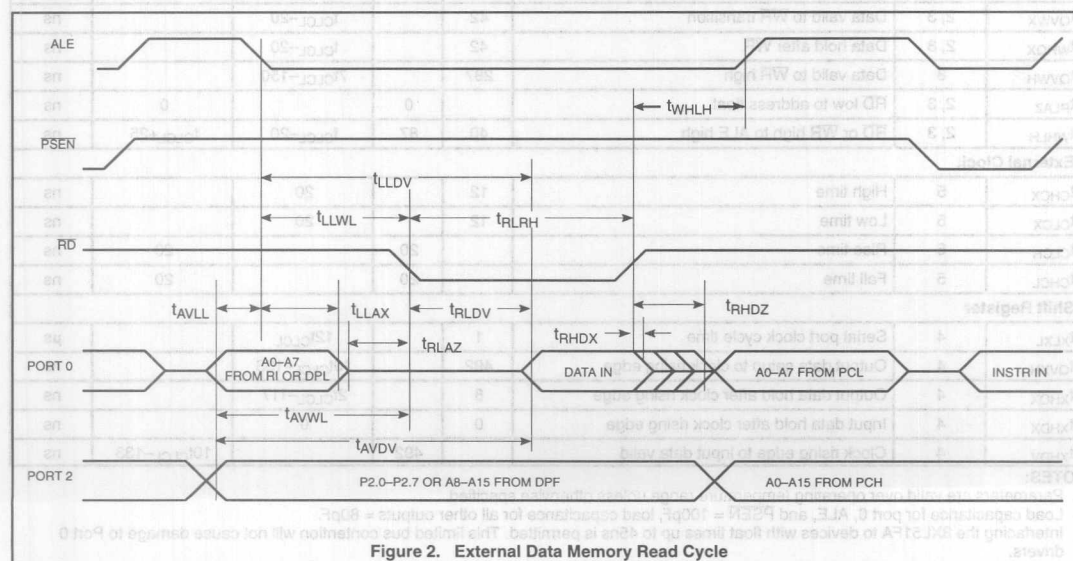


Figure 2. External Data Memory Read Cycle

## CMOS single-chip 3.0V 8-bit microcontroller 83L51FA/87L51FA

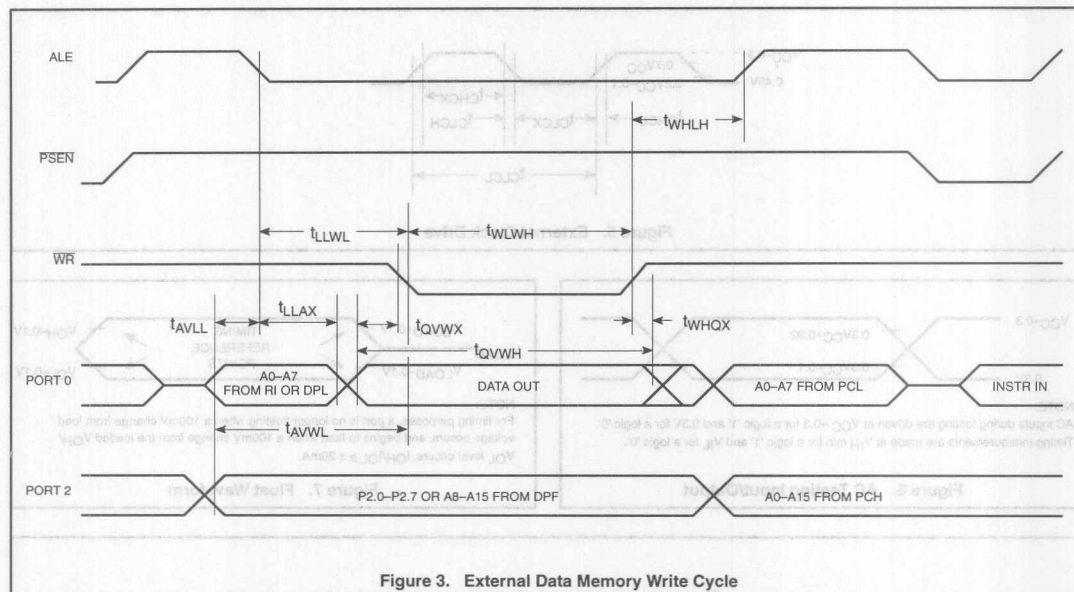


Figure 3. External Data Memory Write Cycle

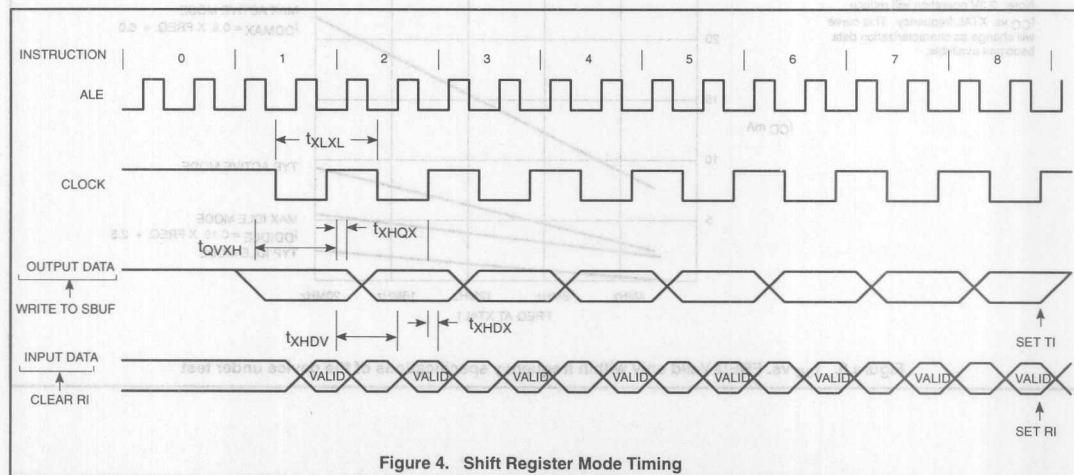
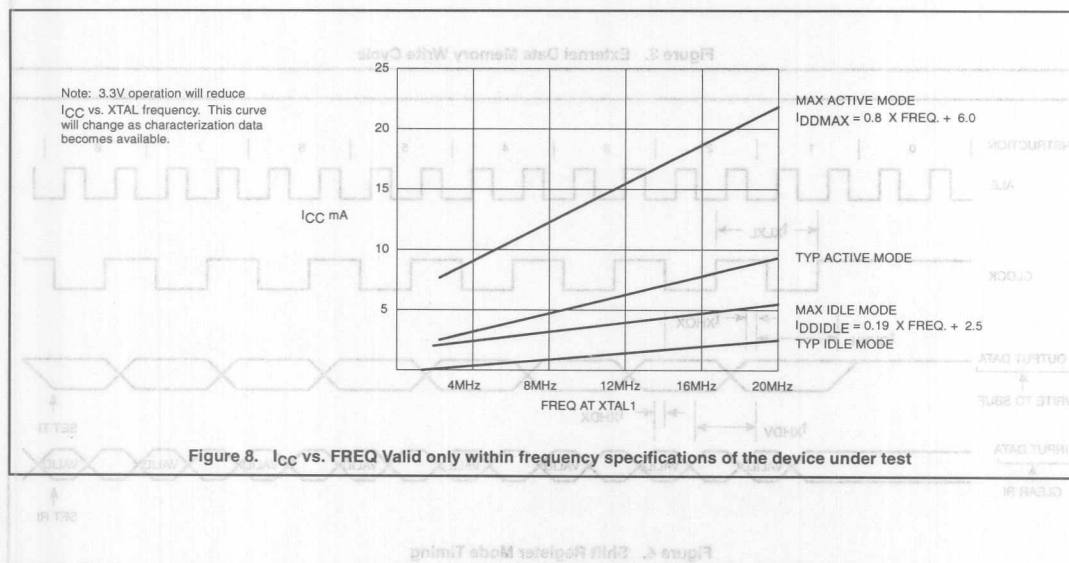
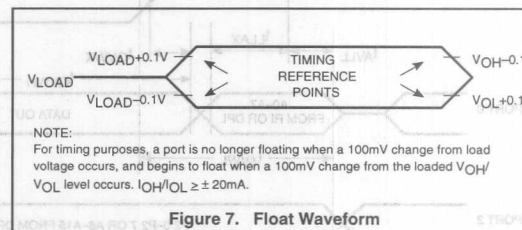
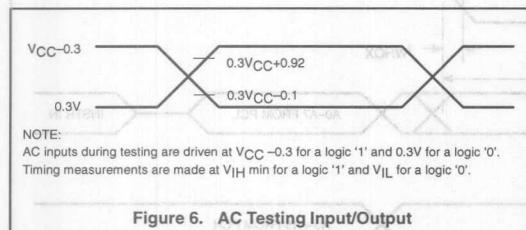
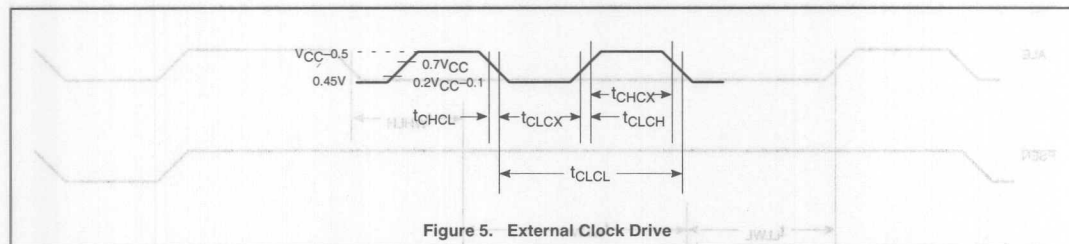


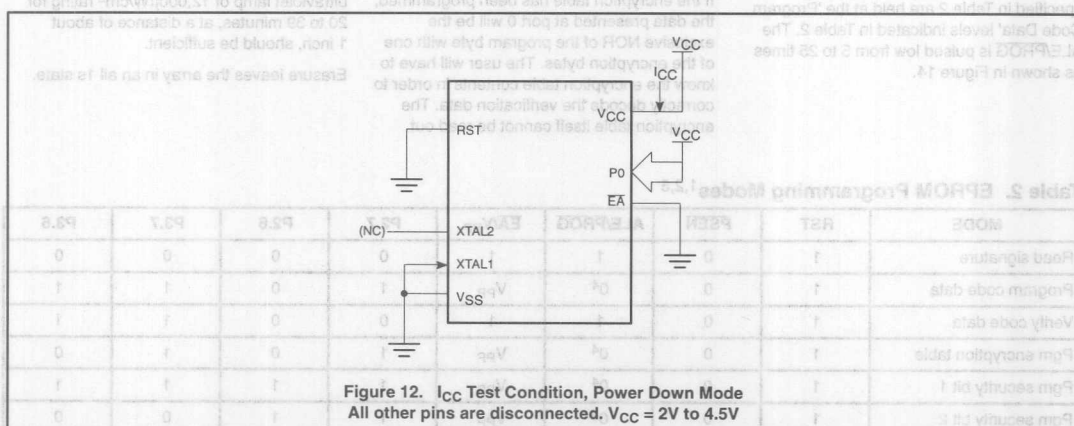
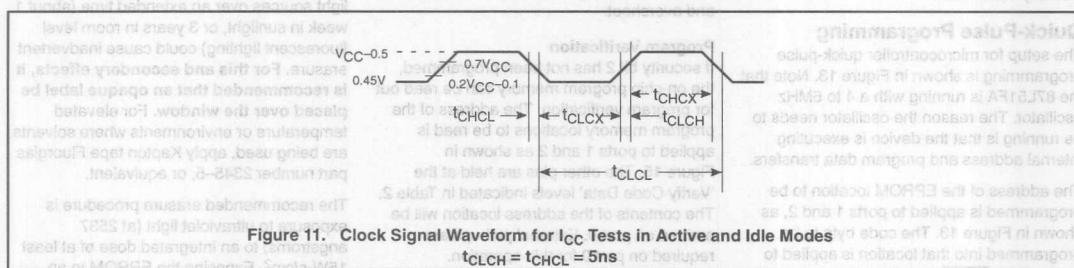
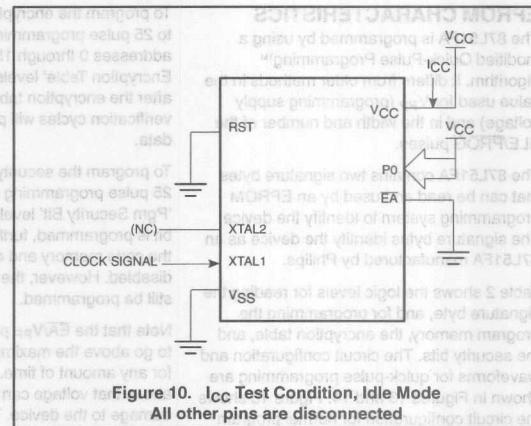
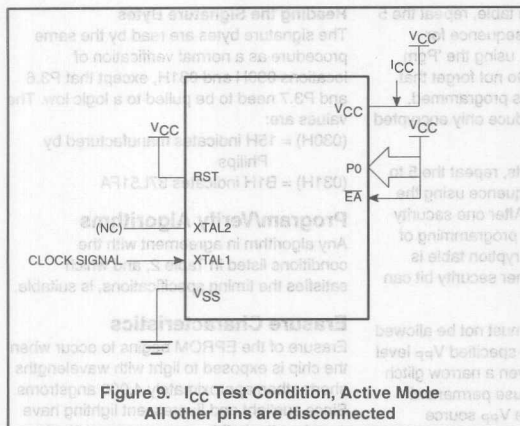
Figure 4. Shift Register Mode Timing

## CMOS single-chip 3.0V 8-bit microcontroller 83L51FA/87L51FA





## CMOS single-chip 3.0V 8-bit microcontroller 83L51FA/87L51FA





## CMOS single-chip 3.0V 8-bit microcontroller 83L51FA/87L51FA

### EPROM CHARACTERISTICS

The 87L51FA is programmed by using a modified Quick-Pulse Programming™ algorithm. It differs from older methods in the value used for  $V_{PP}$  (programming supply voltage) and in the width and number of the ALE/PROG pulses.

The 87L51FA contains two signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes identify the device as an 87L51FA manufactured by Philips.

Table 2 shows the logic levels for reading the signature byte, and for programming the program memory, the encryption table, and the security bits. The circuit configuration and waveforms for quick-pulse programming are shown in Figures 13 and 14. Figure 15 shows the circuit configuration for normal program memory verification.

### Quick-Pulse Programming

The setup for microcontroller quick-pulse programming is shown in Figure 13. Note that the 87L51FA is running with a 4 to 6MHz oscillator. The reason the oscillator needs to be running is that the device is executing internal address and program data transfers.

The address of the EPROM location to be programmed is applied to ports 1 and 2, as shown in Figure 13. The code byte to be programmed into that location is applied to port 0. RST, PSEN and pins of ports 2 and 3 specified in Table 2 are held at the 'Program Code Data' levels indicated in Table 2. The ALE/PROG is pulsed low from 5 to 25 times as shown in Figure 14.

To program the encryption table, repeat the 5 to 25 pulse programming sequence for addresses 0 through 1FH, using the 'Pgm Encryption Table' levels. Do not forget that after the encryption table is programmed, verification cycles will produce only encrypted data.

To program the security bits, repeat the 5 to 25 pulse programming sequence using the 'Pgm Security Bit' levels. After one security bit is programmed, further programming of the code memory and encryption table is disabled. However, the other security bit can still be programmed.

Note that the EA/ $V_{PP}$  pin must not be allowed to go above the maximum specified  $V_{PP}$  level for any amount of time. Even a narrow glitch above that voltage can cause permanent damage to the device. The  $V_{PP}$  source should be well regulated and free of glitches and overshoot.

### Program Verification

If security bit 2 has not been programmed, the on-chip program memory can be read out for program verification. The address of the program memory locations to be read is applied to ports 1 and 2 as shown in Figure 15. The other pins are held at the 'Verify Code Data' levels indicated in Table 2. The contents of the address location will be emitted on port 0. External pull-ups are required on port 0 for this operation.

If the encryption table has been programmed, the data presented at port 0 will be the exclusive NOR of the program byte with one of the encryption bytes. The user will have to know the encryption table contents in order to correctly decode the verification data. The encryption table itself cannot be read out.

### Reading the Signature Bytes

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. The values are:

(030H) = 15H indicates manufactured by Philips  
(031H) = B1H indicates 87L51FA

### Program/Verify Algorithms

Any algorithm in agreement with the conditions listed in Table 2, and which satisfies the timing specifications, is suitable.

### Erase Characteristics

Erase of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent erasure. **For this and secondary effects, it is recommended that an opaque label be placed over the window.** For elevated temperature or environments where solvents are being used, apply Kapton tape Fluorglas part number 2345-5, or equivalent.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 angstroms) to an integrated dose of at least 15W-s/cm<sup>2</sup>. Exposing the EPROM to an ultraviolet lamp of 12,000µW/cm<sup>2</sup> rating for 20 to 39 minutes, at a distance of about 1 inch, should be sufficient.

Erase leaves the array in an all 1s state.

Table 2. EPROM Programming Modes<sup>1,2,3</sup>

MODE	RST	PSEN	ALE/PROG	EA/ $V_{PP}$	P2.7	P2.6	P3.7	P3.6
Read signature	1	0	1	1	0	0	0	0
Program code data	1	0	0 <sup>4</sup>	$V_{PP}$	1	0	1	1
Verify code data	1	0	1	1	0	0	1	1
Pgm encryption table	1	0	0 <sup>4</sup>	$V_{PP}$	1	0	1	0
Pgm security bit 1	1	0	0 <sup>4</sup>	$V_{PP}$	1	1	1	1
Pgm security bit 2	1	0	0 <sup>4</sup>	$V_{PP}$	1	1	0	0

#### NOTES:

- '0' = Valid low for that pin, '1' = valid high for that pin.
- $V_{PP}$  = 12.75V  $\pm$  0.25V.
- $V_{CC}$  = 5V  $\pm$  10% during programming and verification.
- ALE/PROG receives 5 to 25 programming pulses while  $V_{PP}$  is held at 12.75V. Each programming pulse is low for 100µs ( $\pm$ 10µs) and high for a minimum of 10µs.

™Trademark phrase of Intel Corporation.

## CMOS single-chip 3.0V 8-bit microcontroller 87L51FA/87L51FA

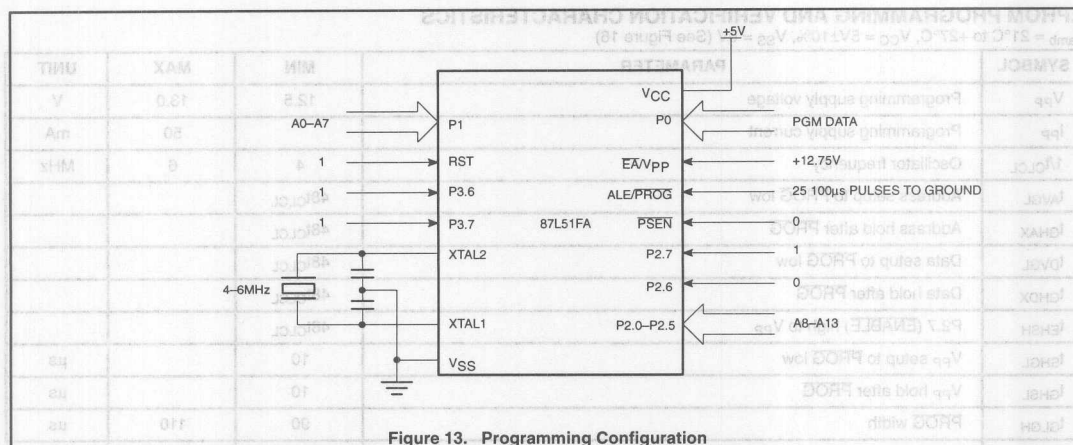


Figure 13. Programming Configuration

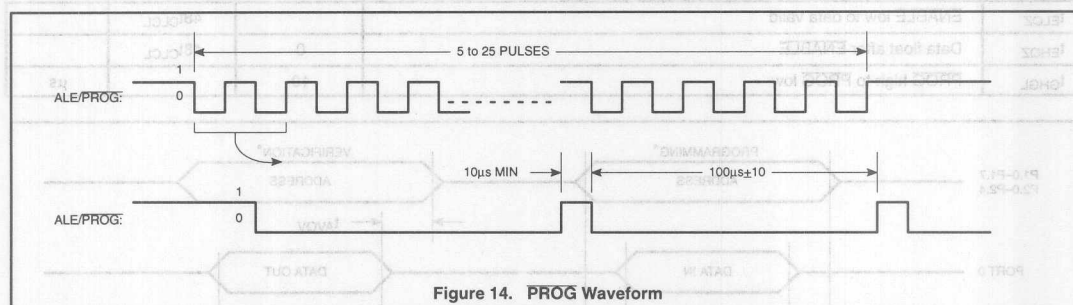


Figure 14. PROG Waveform

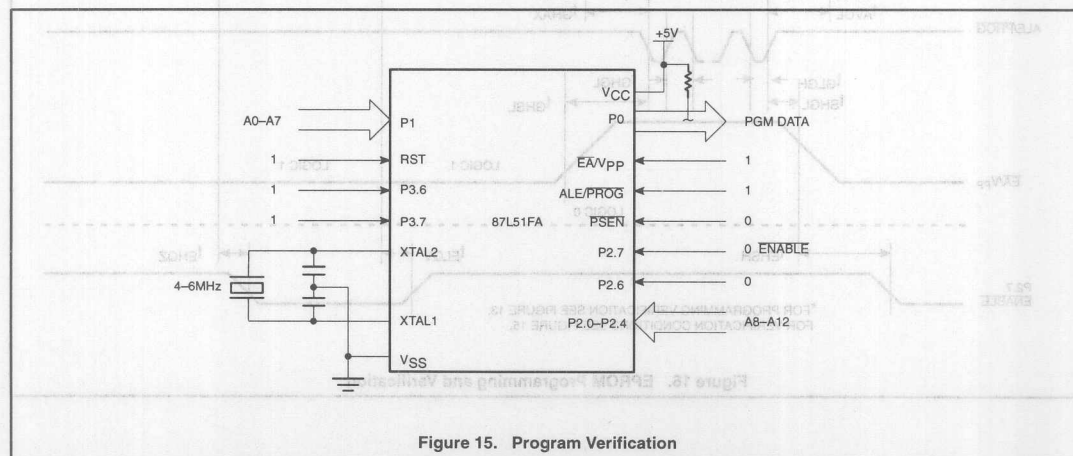


Figure 15. Program Verification

## CMOS single-chip 3.0V 8-bit microcontroller 83L51FA/87L51FA

## EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

 $T_{amb} = 21^{\circ}\text{C}$  to  $+27^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$  (See Figure 16)

SYMBOL	PARAMETER	MIN	MAX	UNIT
$V_{PP}$	Programming supply voltage	12.5	13.0	V
$I_{PP}$	Programming supply current		50	mA
$1/t_{CLCL}$	Oscillator frequency	4	6	MHz
$t_{AVGL}$	Address setup to PROG low	$48t_{CLCL}$		
$t_{GHAX}$	Address hold after PROG	$48t_{CLCL}$		
$t_{DVGL}$	Data setup to PROG low	$48t_{CLCL}$		
$t_{GHDX}$	Data hold after PROG	$48t_{CLCL}$		
$t_{ESH}$	P2.7 (ENABLE) high to $V_{PP}$	$48t_{CLCL}$		
$t_{SHGL}$	$V_{PP}$ setup to PROG low	10		$\mu\text{s}$
$t_{GHSL}$	$V_{PP}$ hold after PROG	10		$\mu\text{s}$
$t_{GLGH}$	PROG width	90	110	$\mu\text{s}$
$t_{AVQV}$	Address to data valid		$48t_{CLCL}$	
$t_{ELOZ}$	ENABLE low to data valid		$48t_{CLCL}$	
$t_{EHQZ}$	Data float after ENABLE	0	$48t_{CLCL}$	
$t_{GHGL}$	PROG high to PROG low	10		$\mu\text{s}$

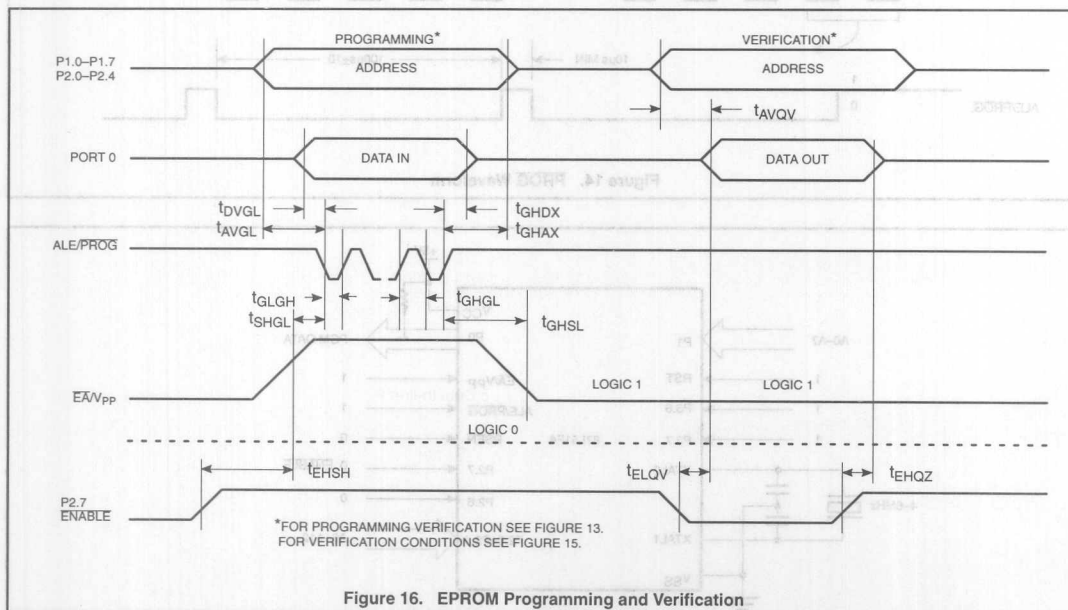


Figure 16. EPROM Programming and Verification

## CMOS single-chip 3.0V 8-bit microcontroller 83L51FB/87L51FB

## DESCRIPTION

The 83L51FB and 87L51FB (hereafter generically referred to as 8XL51FB) Single-Chip 3.0V 8-Bit Microcontrollers are manufactured in an advanced CMOS process and are derivatives of the 80C51 microcontroller family. The 8XL51FB has the same instruction set as the 80C51.

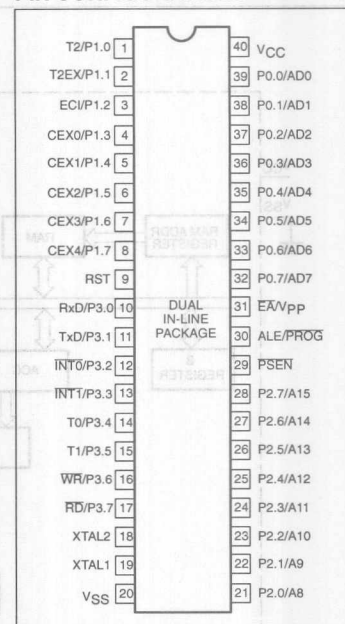
This device provides architectural enhancements that make it applicable in a variety of applications for general control systems. The 87L51FB contains 16k x 8 memory, a volatile 256 x 8 read/write data memory, four 8-bit I/O ports, three 16-bit timer/event counters, a Programmable Counter Array (PCA), a multi-source, two-priority-level, nested interrupt structure, an enhanced UART and on-chip oscillator and timing circuits. For systems that require extra capability, the 8XL51FB can be expanded using standard 3.3V TTL compatible memories and logic.

Its added features make it an even more powerful microcontroller for applications that require pulse width modulation, high-speed I/O and up/down counting capabilities such as motor control. It also has a more versatile serial channel that facilitates multiprocessor communications.

## FEATURES

- 80C51 central processing unit
- 3.0 to 4.5V  $V_{CC}$  range
- 16k x 8 EPROM (87L51FB)  
16k x 8 ROM (83L51FB)
  - expandable externally to 64k bytes
  - Quick Pulse programming algorithm
  - Two level program security system
- 256 x 8 RAM, expandable externally to 64k bytes
- Three 16-bit timer/counters
  - T2 is an up/down counter
- Programmable Counter Array (PCA)
  - High speed output
  - Capture/compare
  - Pulse Width Modulator
  - Watchdog Timer
- Four 8-bit I/O ports
- Full-duplex enhanced UART
  - Framing error detection
  - Automatic address recognition
- Power control modes
  - Idle mode
  - Power-down mode
- Once (On Circuit Emulation) Mode
- Five package styles
- OTP package available

## PIN CONFIGURATIONS



## ORDERING INFORMATION

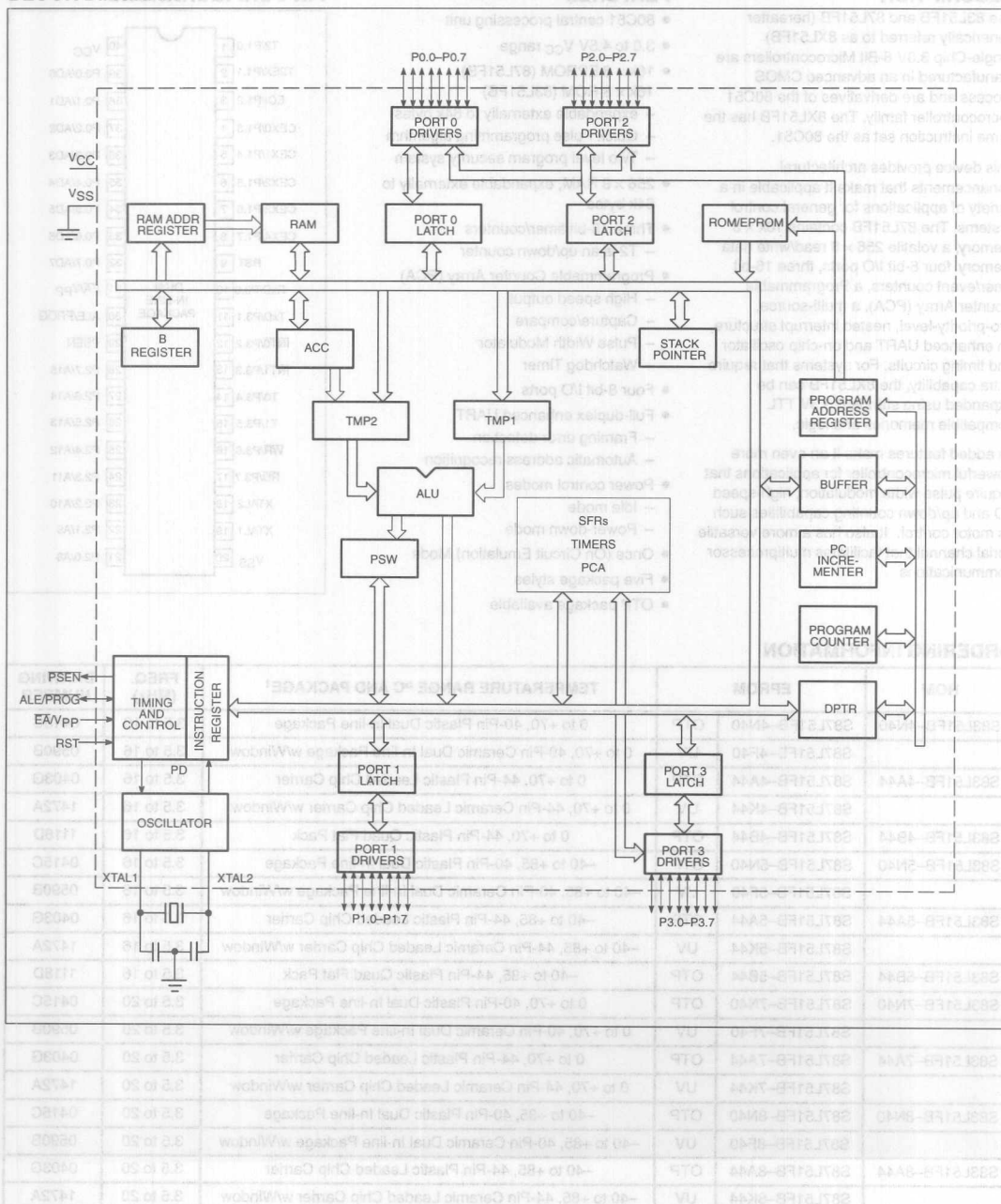
ROM	EPROM		TEMPERATURE RANGE °C AND PACKAGE <sup>1</sup>	FREQ. (MHz)	DRAWING NUMBER
S83L51FB-4N40	S87L51FB-4N40	OTP	0 to +70, 40-Pin Plastic Dual In-line Package	3.5 to 16	0415C
	S87L51FB-4F40	UV	0 to +70, 40-Pin Ceramic Dual In-line Package w/Window	3.5 to 16	0590B
S83L51FB-4A44	S87L51FB-4A44	OTP	0 to +70, 44-Pin Plastic Leaded Chip Carrier	3.5 to 16	0403G
	S87L51FB-4K44	UV	0 to +70, 44-Pin Ceramic Leaded Chip Carrier w/Window	3.5 to 16	1472A
S83L51FB-4B44	S87L51FB-4B44	OTP	0 to +70, 44-Pin Plastic Quad Flat Pack	3.5 to 16	1118D
S83L51FB-5N40	S87L51FB-5N40	OTP	-40 to +85, 40-Pin Plastic Dual In-line Package	3.5 to 16	0415C
	S87L51FB-5F40	UV	-40 to +85, 40-Pin Ceramic Dual In-line Package w/Window	3.5 to 16	0590B
S83L51FB-5A44	S87L51FB-5A44	OTP	-40 to +85, 44-Pin Plastic Leaded Chip Carrier	3.5 to 16	0403G
	S87L51FB-5K44	UV	-40 to +85, 44-Pin Ceramic Leaded Chip Carrier w/Window	3.5 to 16	1472A
S83L51FB-5B44	S87L51FB-5B44	OTP	-40 to +85, 44-Pin Plastic Quad Flat Pack	3.5 to 16	1118D
S83L51FB-7N40	S87L51FB-7N40	OTP	0 to +70, 40-Pin Plastic Dual In-line Package	3.5 to 20	0415C
	S87L51FB-7F40	UV	0 to +70, 40-Pin Ceramic Dual In-line Package w/Window	3.5 to 20	0590B
S83L51FB-7A44	S87L51FB-7A44	OTP	0 to +70, 44-Pin Plastic Leaded Chip Carrier	3.5 to 20	0403G
	S87L51FB-7K44	UV	0 to +70, 44-Pin Ceramic Leaded Chip Carrier w/Window	3.5 to 20	1472A
S83L51FB-8N40	S87L51FB-8N40	OTP	-40 to +85, 40-Pin Plastic Dual In-line Package	3.5 to 20	0415C
	S87L51FB-8F40	UV	-40 to +85, 40-Pin Ceramic Dual In-line Package w/Window	3.5 to 20	0590B
S83L51FB-8A44	S87L51FB-8A44	OTP	-40 to +85, 44-Pin Plastic Leaded Chip Carrier	3.5 to 20	0403G
	S87L51FB-8K44	UV	-40 to +85, 44-Pin Ceramic Leaded Chip Carrier w/Window	3.5 to 20	1472A

## NOTE:

1. OTP = One Time Programmable EPROM. UV = Erasable EPROM.

## CMOS single-chip 3.0V 8-bit microcontroller 83L51FB/87L51FB

### BLOCK DIAGRAM

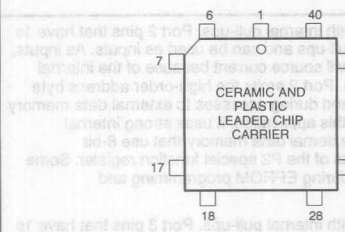




## CMOS single-chip 3.0V 8-bit microcontroller 83L51FB/87L51FB

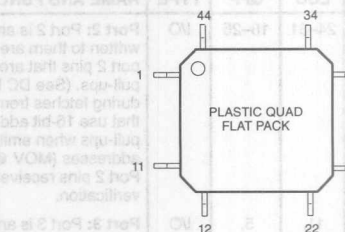
## CERAMIC AND PLASTIC LEADED CHIP CARRIER

## PIN FUNCTIONS



Pin	Function	Pin	Function
1	NC	23	NC
2	P1.0/T2	24	P2.0/A8
3	P1.1/T2EX	25	P2.1/A9
4	P1.2/ECI	26	P2.2/A10
5	P1.3/CEX0	27	P2.3/A11
6	P1.4/CEX1	28	P2.4/A12
7	P1.5/CEX2	29	P2.5/A13
8	P1.6/CEX3	30	P2.6/A14
9	P1.7/CEX4	31	P2.7/A15
10	RST	32	PSEN
11	P3.0/RxD	33	ALE/PROG
12	NC	34	NC
13	P3.1/TxD	35	EA/Vpp
14	P3.2/INT0	36	P0.7/AD7
15	P3.3/INT1	37	P0.6/AD6
16	P3.4/T0	38	P0.5/AD5
17	P3.5/T1	39	P0.4/AD4
18	P3.6/WR	40	P0.3/AD3
19	P3.7/RD	41	P0.2/AD2
20	XTAL2	42	P0.1/AD1
21	XTAL1	43	P0.0/AD0
22	VSS	44	VCC

## PLASTIC QUAD FLAT PACK PIN FUNCTIONS



Pin	Function	Pin	Function
1	P1.5/CEX2	23	P2.5/A13
2	P1.6/CEX3	24	P2.6/A14
3	P1.7/CEX4	25	P2.7/A15
4	RST	26	PSEN
5	P3.0/RxD	27	ALE/PROG
6	NC	28	NC
7	P3.1/TxD	29	EA/Vpp
8	P3.2/INT0	30	P0.7/AD7
9	P3.3/INT1	31	P0.6/AD6
10	P3.4/T0	32	P0.5/AD5
11	P3.5/T1	33	P0.4/AD4
12	P3.6/WR	34	P0.3/AD3
13	P3.7/RD	35	P0.2/AD2
14	XTAL2	36	P0.1/AD1
15	XTAL1	37	P0.0/AD0
16	VSS	38	VCC
17	NC	39	NC
18	P2.0/A8	40	P1.0/T2
19	P2.1/A9	41	P1.1/T2EX
20	P2.2/A10	42	P1.2/ECI
21	P2.3/A11	43	P1.3/CEX0
22	P2.4/A12	44	P1.4/CEX1

## PIN DESCRIPTIONS

MNEMONIC	PIN NUMBER			TYPE	NAME AND FUNCTION
DIP	LCC	QFP			
VSS	20	22	16	I	Ground: 0V reference.
VCC	40	44	38	I	<b>Power Supply:</b> This is the power supply voltage for normal, idle, and power-down operation.
P0.0–0.7	39–32	43–36	37–30	I/O	<b>Port 0:</b> Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s. Port 0 also outputs the code bytes during program verification and receives code bytes during EPROM programming. External pull-ups are required during program verification.
P1.0–P1.7	1–8	2–9	40–44, 1–3	I/O	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pull-ups, except P1.6 and P1.7 which are open drain. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: $I_{IL}$ ). Port 1 also receives the low-order address byte during program memory verification. Alternate functions include: <b>T2 (P1.0):</b> Timer/Counter 2 external count input/Clockout <b>T2EX (P1.1):</b> Timer/Counter 2 Reload/Capture/Direction Control <b>ECI (P1.2):</b> External Clock Input to the PCA <b>CEX0 (P1.3):</b> Capture/Compare External I/O for PCA module 0 <b>CEX1 (P1.4):</b> Capture/Compare External I/O for PCA module 1 <b>CEX2 (P1.5):</b> Capture/Compare External I/O for PCA module 2 <b>CEX3 (P1.6):</b> Capture/Compare External I/O for PCA module 3 <b>CEX4 (P1.7):</b> Capture/Compare External I/O for PCA module 4
	1	2	40	I	
	2	3	41	I	
	3	4	42	I	
	4	5	43	I/O	
	5	6	44	I/O	
	6	7	1	I/O	
	7	8	2	I/O	
	8	9	3	I/O	



## CMOS single-chip 3.0V 8-bit microcontroller 883L51FB/87L51FB

## PIN DESCRIPTIONS (Continued)

MNEMONIC	PIN NUMBER			TYPE	NAME AND FUNCTION
	DIP	LCC	QFP		
P2.0–P2.7	21–28	24–31	18–25	I/O	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: $I_{IL}$ ). Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register. Some Port 2 pins receive the high order address bits during EPROM programming and verification.
P3.0–P3.7	10–17	11, 13–19	5, 7–13	I/O	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: $I_{IL}$ ). Port 3 also serves the special features of the 80C51 family, as listed below: <b>RxD (P3.0):</b> Serial input port <b>TxD (P3.1):</b> Serial output port <b>INT0 (P3.2):</b> External interrupt <b>INT1 (P3.3):</b> External interrupt <b>T0 (P3.4):</b> Timer 0 external input <b>T1 (P3.5):</b> Timer 1 external input <b>WR (P3.6):</b> External data memory write strobe <b>RD (P3.7):</b> External data memory read strobe
RST	9	10	4	I	<b>Reset:</b> A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to $V_{SS}$ permits a power-on reset using only an external capacitor to $V_{CC}$ .
ALE/PROG	30	33	27	I/O	<b>Address Latch Enable/Program Pulse:</b> Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory. This pin is also the program pulse input (PROG) during EPROM programming.
PSEN	29	32	26	O	<b>Program Store Enable:</b> The read strobe to external program memory. When the 88L51FB is executing code from the external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory. PSEN is not activated during fetches from internal program memory.
EA/ $V_{PP}$	31	35	29	I	<b>External Access Enable/Programming Supply Voltage:</b> EA must be externally held low to enable the device to fetch code from external program memory locations 0000H and 3FFFFH. If EA is held high, the device executes from internal program memory unless the program counter contains an address greater than 3FFFFH. This pin also receives the 12.75V programming supply voltage ( $V_{PP}$ ) during EPROM programming. If security bit 1 is programmed, EA will be internally latched on Reset.
XTAL1	19	21	15	I	<b>Crystal 1:</b> Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	20	14	O	<b>Crystal 2:</b> Output from the inverting oscillator amplifier.

## NOTE:

To avoid "latch-up" effect at power-on, the voltage on any pin at any time must not be higher than  $V_{CC} + 0.5V$  or  $V_{SS} - 0.5V$ , respectively.

## TIMER 2

This is a 16-bit up or down counter, which can be operated as either a timer or event counter. It can be operated in one of three different modes (autoreload, capture or as the baud rate generator for the UART).

In the autoreload mode the Timer can be set to count up or down by setting or clearing the bit DCEN in the T2CON Special Function Register. The SFR's RCAP2H and RCAP2L are used to reload the Timer upon overflow or a 1-to-0 transition on the T2EX input (P1.1).

In the Capture mode Timer 2 can either set TF2 and generate an interrupt or capture its value. To capture Timer 2 in response to a 1-to-0 transition on the T2EX input, the EXEN2 bit in the T2CON must be set. Timer 2 is then captured in SFR's RCAP2H and RCAP2L.

As the baud rate generator, Timer 2 is selected by setting TCLK and/or RCLK in T2CON. As the baud rate generator Timer 2 is incremented at  $1/2$  the oscillator frequency.

## ENHANCED UART

The 88L51FB UART has all of the capabilities of the standard 80C51 UART plus Framing Error Detection and Automatic Address Recognition. As in the 80C51, all four modes of operation are supported as well as the 9th bit in modes 2 and 3 that can be used to facilitate multiprocessor communication.

The Framing Error Detection allows the UART to look for missing stop bits. If a Stop

## CMOS single-chip 3.0V 8-bit microcontroller 83L51FB/87L51FB

bit is missing, the FE bit in the SCON SFR is set. The FE bit can be checked after each transmission to detect communication errors. The FE bit can only be cleared by software and is not affected by a valid stop bit.

Automatic Address Recognition is used to reduce the CPU service time for the serial port. The CPU only needs to service the UART when it is addressed and, with this done by the on-chip circuitry, the need for software overhead is greatly reduced. This mode works similar to the 9-bit communication mode, except that it uses only 8 bits and the Stop bit is used to cause the RI bit to be set. There are two SFRs associated with this mode. They are SADDR, which holds the slave address and SADEN, which contains a mask that allows selective masking of the slave address so that broadcast addresses can be used.

### PROGRAMMABLE COUNTER ARRAY

The PCA is a sophisticated free-running 16 bit Timer/Counter that drives 5 modules that can be individually configured as Capture inputs, software timers, high speed outputs, or pulse width modulated outputs. In addition, module 4 can be configured as a software controlled watchdog timer.

The Timer portion of the PCA can be configured to run in one of four different modes. The modes are:  $1/2$  the oscillator frequency,  $1/4$  the oscillator frequency, Timer 0 overflows, or from the ECI input.

For the Capture/Compare mode each of the modules has a pair of registers associated with it called CCAPnH and CCAPnL (where  $n = 0, 1, 2, 3, 4$  depending on the module). Both positive and negative transitions can be captured. This means that the PCA has the flexibility to measure phase differences, duty cycles, pulse widths and a wide variety of other digital pulse characteristics.

In the 16-bit software timer mode each of the modules can generate an interrupt upon a compare.

For applications that require accurate pulse widths and edges the PCA modules can be used as High Speed Outputs (HSO). The PCA toggles the appropriate CEXn pin when there is a match between the PCA timer and the modules compare registers.

The pulse width modulator mode for the PCA allows the conversion of digital information into analog signals. Each of the 5 modules can be used in this mode. The frequency of the PWM depends on the clock source for the PCA. The 8-bit PWM output is generated

by comparing the low byte of the PCA (CL) with the module's CCAPnL SFR. When  $CL < CCAPnL$ , the output is high. When  $CL > CCAPnL$ , the output is low.

### POWER OFF FLAG

The Power Off Flag (POF) is set by on-chip circuitry when the  $V_{CC}$  level on the 83L51FB rises from 0 to 3.3V. The POF bit can be set or cleared by software allowing a user to determine if the reset is the result of a power-on or a warm start after powerdown. The  $V_{CC}$  level must remain above 2.0V for the POF to remain unaffected by the  $V_{CC}$  level.

### OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

### Reset

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. To insure a good power-on reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few milliseconds) plus two machine cycles. At power-on, the voltage on  $V_{CC}$  and RST must come up at the same time for a proper start-up.

### Idle Mode

In the idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

### Power-Down Mode

To save even more power, a Power Down mode can be invoked by software. In this mode, the oscillator is stopped and the instruction that invoked Power Down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power Down mode is terminated.

On the 83L51FB either a hardware reset or external interrupt can use an exit from Power Down. Reset redefines all the SFRs but does not change the on-chip RAM. An external interrupt allows both the SFRs and the on-chip RAM to retain their values.

To properly terminate Power Down the reset or external interrupt should not be executed before  $V_{CC}$  is restored to its normal operating level and must be held active long enough for the oscillator to restart and stabilize (normally less than 10ms).

With an external interrupt, INT0 and INT1 must be enabled and configured as level-sensitive. Holding the pin low restarts the oscillator but bringing the pin back high completes the exit. Once the interrupt is serviced, the next instruction to be executed after RETI will be the one following the instruction that put the device into Power Down.

### Design Consideration

- When the idle mode is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

### ONCE™ Mode

The ONCE ("On-Circuit Emulation") Mode facilitates testing and debugging of systems using the 83L51FB without the 83L51FB having to be removed from the circuit. The ONCE Mode is invoked by:

1. Pull ALE low while the device is in reset and PSEN is high;
2. Hold ALE low as RST is deactivated.

While the device is in ONCE Mode, the Port 0 pins go into a float state, and the other port pins and ALE and PSEN are weakly pulled high. The oscillator circuit remains active. While the 83L51FB is in this mode, an emulator or test CPU can be used to drive the circuit. Normal operation is restored when a normal reset is applied.

CMOS single-chip 3.0V 8-bit microcontroller 83L51FB/87L51FB

Table 1. External Pin Status During Idle and Power-Down Mode

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>

PARAMETER	RATING	UNIT
Operating temperature under bias	0 to +70 or -40 to +85	°C
Storage temperature range	-65 to +150	°C
Voltage on EA/V <sub>PP</sub> pin to V <sub>SS</sub>	0 to +13.0	V
Voltage on any other pin to V <sub>SS</sub>	-0.5 to +6.5	V
Maximum I <sub>OL</sub> per I/O pin	15	mA
Power dissipation (based on package heat transfer limitations, not device power consumption)	1.5	W

NOTES:

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
- This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
- Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V<sub>SS</sub> unless otherwise noted.

Electrical Deviations from Commercial Specifications for Extended Temperature Range

DC and AC parameters not included here are the same as in the commercial temperature range table.

## CMOS single-chip 3.0V 8-bit microcontroller 83L51FB/87L51FB

## DC ELECTRICAL CHARACTERISTICS

T<sub>amb</sub> = 0°C to +70°C, -40 to +85°C, V<sub>CC</sub> = 3.0V to 4.5V, V<sub>SS</sub> = 0V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP <sup>1</sup>	MAX	
V <sub>IL</sub>	Input low voltage, except EA <sup>7,3</sup>		-0.5		0.8	V
V <sub>IL1</sub>	Input low voltage to EA <sup>7,3</sup>		0		0.8	V
V <sub>IH</sub>	Input high voltage, except XTAL1, RST <sup>7,4</sup>		2.0		V <sub>CC</sub> +0.5	V
V <sub>IH1</sub>	Input high voltage, XTAL1, RST <sup>7,4</sup>		0.7V <sub>CC</sub>		V <sub>CC</sub> +0.5	V
V <sub>OL</sub>	Output low voltage, ports 1, 2, 3 <sup>9</sup>	I <sub>OL</sub> = 1.6mA <sup>2</sup>			0.45	V
V <sub>OL1</sub>	Output low voltage, port 0, ALE, PSEN <sup>9</sup>	I <sub>OL</sub> = 3.2mA <sup>2</sup>			0.45	V
V <sub>OH</sub>	Output high voltage, ports 1, 2, 3, ALE, PSEN <sup>3</sup>	I <sub>OH</sub> = -20µA	V <sub>CC</sub> - 0.5			V
V <sub>OH1</sub>	Output high voltage (port 0 in external bus mode), ALE <sup>10</sup> , PSEN <sup>3</sup>	I <sub>OH</sub> = -3.2mA	V <sub>CC</sub> - 0.7			V
I <sub>IL</sub>	Logical 0 input current, ports 1, 2, 3 <sup>7</sup>	V <sub>IN</sub> = 0.45V			-50	µA
I <sub>TL</sub>	Logical 1-to-0 transition current, ports 1, 2, 3 <sup>7</sup>	See note 4			-650	µA
I <sub>LI</sub>	Input leakage current, port 0	0.45 V <sub>IN</sub> < V <sub>CC</sub> - 0.3			±5	µA
I <sub>CC</sub>	Power supply current: Active mode @ 20MHz <sup>5</sup> Idle mode @ 20MHz Power-down mode	See note 6		9 2 10	22 6 50	mA mA µA
R <sub>RST</sub>	Internal reset pull-down resistor		40		225	kΩ
C <sub>IO</sub>	Pin capacitance <sup>11</sup> (except EA)				15	pF

## NOTES:

- Typical ratings are not guaranteed. The values listed are at room temperature, 3.3V.
- These values apply only to T<sub>amb</sub> = 0°C to +70°C.
- For V<sub>CC</sub> voltages above 3.6V and less than 5.5V, V<sub>IL</sub> = 0.3V<sub>CC</sub> - 0.1
- For V<sub>CC</sub> voltages above 3.6V and less than 5.5V, V<sub>IH</sub> = 0.3V<sub>CC</sub> + .92
- Under steady state (non-transient) conditions, I<sub>OL</sub> must be externally limited as follows:  
Maximum I<sub>OL</sub> per port pin: 15mA (\*NOTE: This is 85°C specification.)  
Maximum I<sub>OL</sub> per 8-bit port: 26mA  
Maximum total I<sub>OL</sub> for all outputs: 71mA  
If I<sub>OL</sub> exceeds the test condition, V<sub>OL</sub> may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
- Capacitive loading on ports 0 and 2 may cause spurious noise to be superimposed on the V<sub>OL</sub>s of ALE and ports 1 and 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100pF), the noise pulse on the ALE pin may exceed 0.8V. In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input. I<sub>OL</sub> can exceed these conditions provided that no single output sinks more than 5mA and no more than two outputs exceed the test conditions.
- Capacitive loading on ports 0 and 2 may cause the V<sub>OH</sub> on ALE and PSEN to momentarily fall below the 0.9V<sub>CC</sub> specification when the address bits are stabilizing.
- ALE is tested to V<sub>OH1</sub>, except when ALE is off then V<sub>OH</sub> is the voltage specification.
- Pins of ports 1, 2 and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when V<sub>IN</sub> is approximately 1.5V.
- I<sub>CCMAX</sub> at other frequencies is given by: Active mode: I<sub>CCMAX</sub> = 0.8 × FREQ + 6; Idle mode: I<sub>CCMAX</sub> = 0.19 × FREQ + 2.50, where FREQ is the external oscillator frequency in MHz. I<sub>CCMAX</sub> is given in mA. See Figure 23.
- See Figures 24 through 27 for I<sub>CC</sub> test conditions.
- Pin capacitance is less than 25pF. Pin capacitance of ceramic package is less than 15pF (except EA it is 25pF). These values are guaranteed by design and are not tested.



## CMOS single-chip 3.0V 8-bit microcontroller 83L51FB/87L51FB

## AC ELECTRICAL CHARACTERISTICS

 $T_{amb} = 0^{\circ}\text{C to } +70^{\circ}\text{C, } -40 \text{ to } +85^{\circ}\text{C, } V_{CC} = 3.0\text{V to } 4.5\text{V, } V_{SS} = 0\text{V}^{1, 2, 3}$ 

SYMBOL	FIGURE	PARAMETER	16MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
$1/t_{CLCL}$	16	Oscillator frequency			3.5	16	MHz
					3.5	20	MHz
$t_{LHLL}$	16	ALE pulse width	85		$2t_{CLCL}-40$		ns
$t_{AVLL}$	16	Address valid to ALE low	22		$t_{CLCL}-40$		ns
$t_{LLAX}$	16	Address hold after ALE low	32		$t_{CLCL}-30$		ns
$t_{LLIV}$	16	ALE low to valid instruction in		150	$4t_{CLCL}-100$		ns
$t_{LLPL}$	16	ALE low to PSEN low	32		$t_{CLCL}-30$		ns
$t_{PLPH}$	16	PSEN pulse width	142		$3t_{CLCL}-45$		ns
$t_{PLIV}$	16	PSEN low to valid instruction in		82	$3t_{CLCL}-105$		ns
$t_{PXIX}$	16	Input instruction hold after PSEN	0		0		ns
$t_{PXIZ}$	16	Input instruction float after PSEN		37	$t_{CLCL}-25$		ns
$t_{AVIV}$	16	Address to valid instruction in		207	$5t_{CLCL}-105$		ns
$t_{PLAZ}$	16	PSEN low to address float		10	10		ns
<b>Data Memory</b>							
$t_{RLRH}$	17, 18	RD pulse width	275		$6t_{CLCL}-100$		ns
$t_{WLWH}$	17, 18	WR pulse width	275		$6t_{CLCL}-100$		ns
$t_{RLDV}$	17, 18	RD low to valid data in		147	$5t_{CLCL}-165$		ns
$t_{RHDX}$	17, 18	Data hold after RD	0		0		ns
$t_{RHDZ}$	17, 18	Data float after RD		65	$2t_{CLCL}-60$		ns
$t_{LLDV}$	17, 18	ALE low to valid data in		350	$8t_{CLCL}-150$		ns
$t_{AVDV}$	17, 18	Address to valid data in		397	$9t_{CLCL}-165$		ns
$t_{LLWL}$	17, 18	ALE low to RD or WR low	137	237	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
$t_{AVWL}$	17, 18	Address valid to WR low or RD low	175		$4t_{CLCL}-75$		ns
$t_{QVWX}$	17, 18	Data valid to WR transition	42		$t_{CLCL}-20$		ns
$t_{WHQX}$	17, 18	Data hold after WR	42		$t_{CLCL}-20$		ns
$t_{QVWH}$	18	Data valid to WR high	287		$7t_{CLCL}-150$		ns
$t_{RLAZ}$	17, 18	RD low to address float		0	0		ns
$t_{WHLH}$	17, 18	RD or WR high to ALE high	40	87	$t_{CLCL}-20$	$t_{CLCL}+25$	ns
<b>External Clock</b>							
$t_{CHCX}$	20	High time	12		20		ns
$t_{CLCX}$	20	Low time	12		20		ns
$t_{CLCH}$	20	Rise time		20		20	ns
$t_{CHCL}$	20	Fall time		20		20	ns
<b>Shift Register</b>							
$t_{XLXL}$	19	Serial port clock cycle time	1		$12t_{CLCL}$		$\mu\text{s}$
$t_{QVXH}$	19	Output data setup to clock rising edge	492		$10t_{CLCL}-133$		ns
$t_{XHDX}$	19	Output data hold after clock rising edge	8		$2t_{CLCL}-117$		ns
$t_{XHDX}$	19	Input data hold after clock rising edge	0		0		ns
$t_{XHDV}$	19	Clock rising edge to input data valid		492		$10t_{CLCL}-133$	ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- Interfacing the 83L51FB to devices with float times up to 45ns is permitted. This limited bus contention will not cause damage to Port 0 drivers.

## 83L51FB/87L51FB

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

P - PSEN  
Q - Output data  
R - RD signal  
t - Time  
V - Valid  
W - WR signal  
X - No longer a valid logic level  
Z - Float

**Examples:**  $t_{AVLL}$  = Time for address valid to ALE low.  $t_{LLPL}$  = Time for ALE low to PSEN low.





## CMOS single-chip 3.0V 8-bit microcontroller 83L51FB/87L51FB

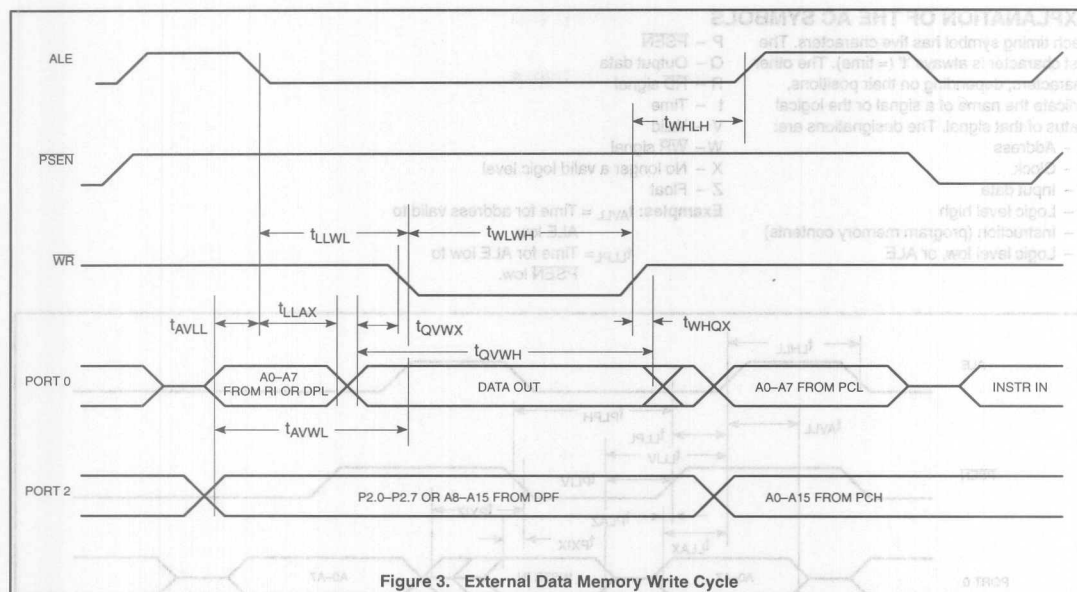


Figure 3. External Data Memory Write Cycle

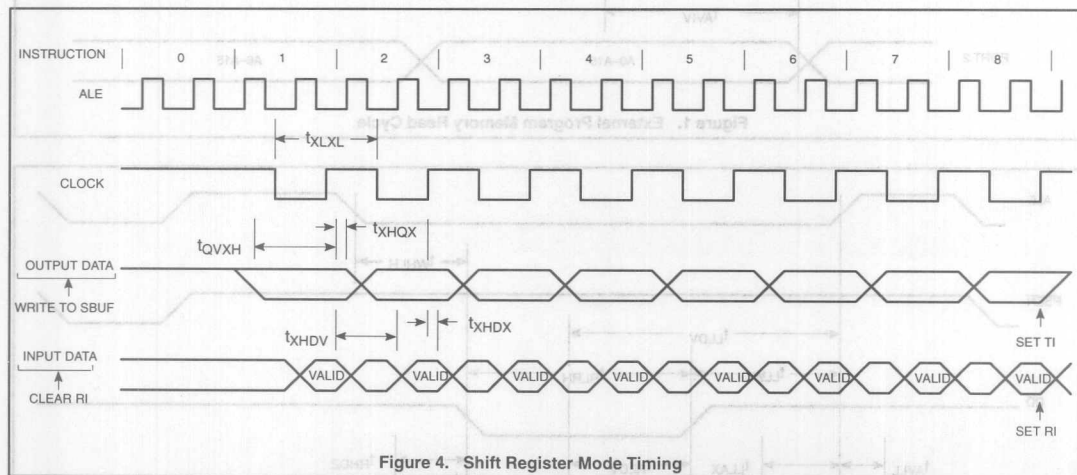
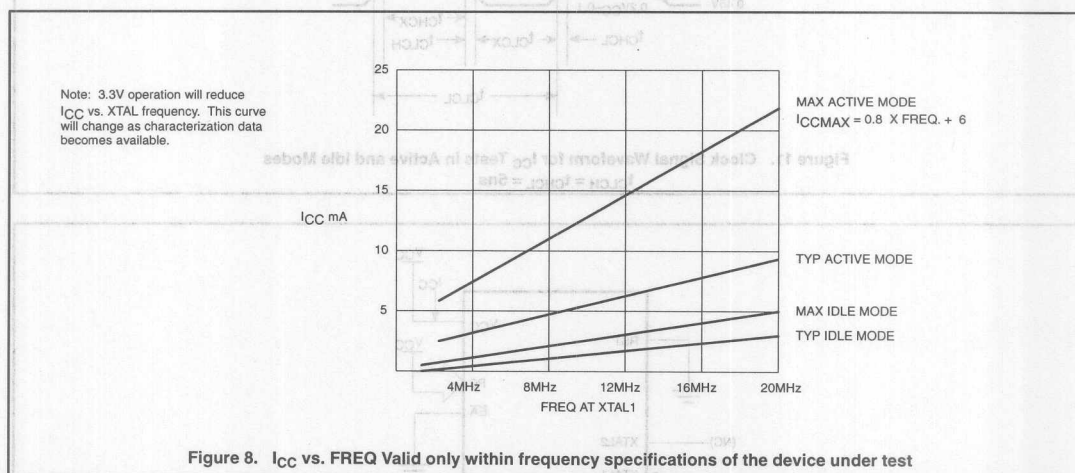
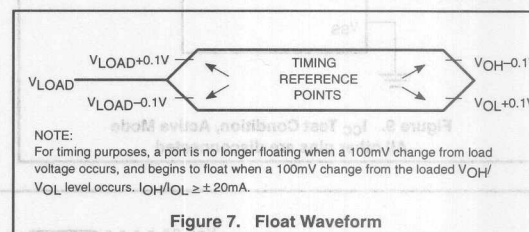
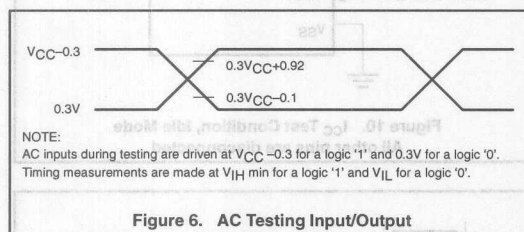
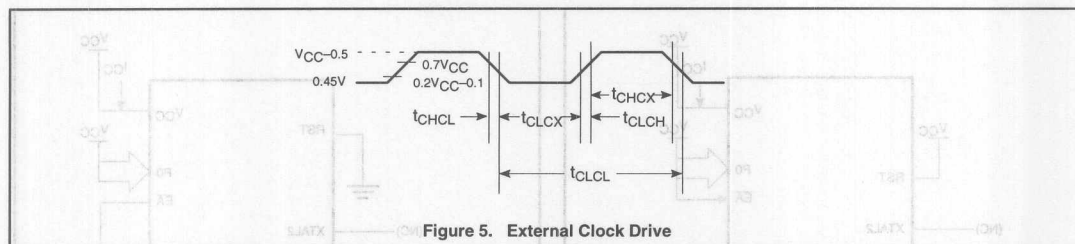
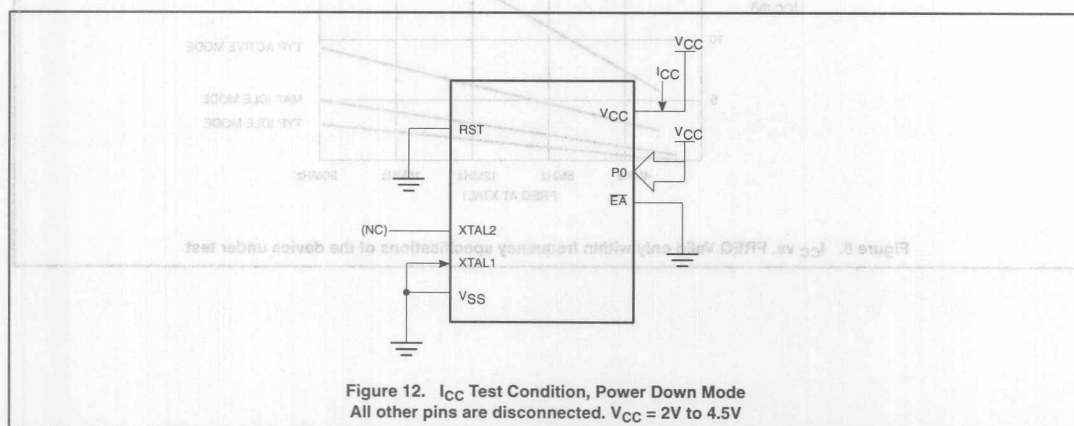
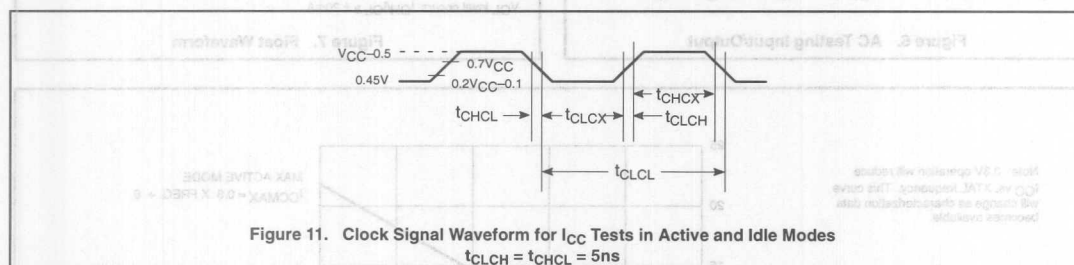
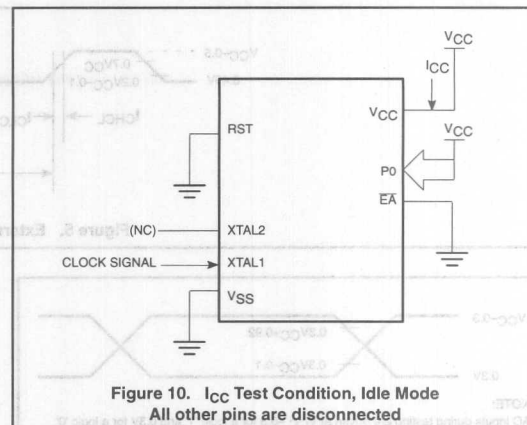
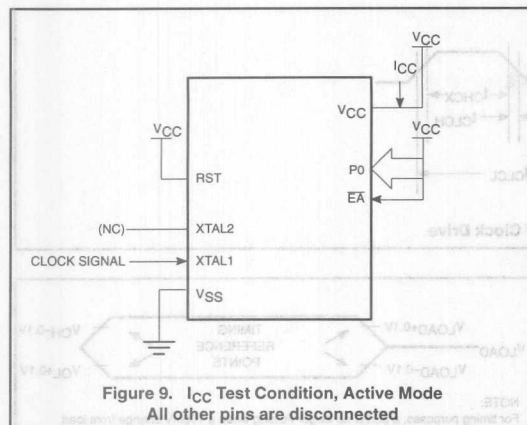


Figure 4. Shift Register Mode Timing

## CMOS single-chip 3.0V 8-bit microcontroller 83L51FB/87L51FB



## CMOS single-chip 3.0V 8-bit microcontroller 83L51FB/87L51FB



## CMOS single-chip 3.0V 8-bit microcontroller 83L51FB/87L51FB

## EPROM CHARACTERISTICS

The 87L51FB is programmed by using a modified Quick-Pulse Programming™ algorithm. It differs from older methods in the value used for  $V_{PP}$  (programming supply voltage) and in the width and number of the ALE/PROG pulses.

The 87L51FB contains two signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes identify the device as an 87L51FB manufactured by Philips.

Table 3 shows the logic levels for reading the signature byte, and for programming the program memory, the encryption table, and the security bits. The circuit configuration and waveforms for quick-pulse programming are shown in Figures 28 and 29. Figure 30 shows the circuit configuration for normal program memory verification.

## Quick-Pulse Programming

The setup for microcontroller quick-pulse programming is shown in Figure 28. Note that the 87L51FB is running with a 4 to 6MHz oscillator. The reason the oscillator needs to be running is that the device is executing internal address and program data transfers.

The address of the EPROM location to be programmed is applied to ports 1 and 2, as shown in Figure 28. The code byte to be programmed into that location is applied to port 0. RST, PSEN and pins of ports 2 and 3 specified in Table 3 are held at the 'Program Code Data' levels indicated in Table 3. The ALE/PROG is pulsed low from 5 to 25 times as shown in Figure 29.

To program the encryption table, repeat the 5 to 25 pulse programming sequence for addresses 0 through 1FH, using the 'Pgm Encryption Table' levels. Do not forget that after the encryption table is programmed, verification cycles will produce only encrypted data.

To program the security bits, repeat the 5 to 25 pulse programming sequence using the 'Pgm Security Bit' levels. After one security bit is programmed, further programming of the code memory and encryption table is disabled. However, the other security bit can still be programmed.

Note that the  $\overline{EA}/V_{PP}$  pin must not be allowed to go above the maximum specified  $V_{PP}$  level for any amount of time. Even a narrow glitch above that voltage can cause permanent damage to the device. The  $V_{PP}$  source should be well regulated and free of glitches and overshoot.

## Program Verification

If security bit 2 has not been programmed, the on-chip program memory can be read out for program verification. The address of the program memory locations to be read is applied to ports 1 and 2 as shown in Figure 30. The other pins are held at the 'Verify Code Data' levels indicated in Table 3. The contents of the address location will be emitted on port 0. External pull-ups are required on port 0 for this operation.

If the encryption table has been programmed, the data presented at port 0 will be the exclusive NOR of the program byte with one of the encryption bytes. The user will have to know the encryption table contents in order to correctly decode the verification data. The encryption table itself cannot be read out.

## Reading the Signature Bytes

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. The values are:

(030H) = 15H indicates manufactured by Philips

(031H) = B2H indicates 87L51FB

## Program/Verify Algorithms

Any algorithm in agreement with the conditions listed in Table 3, and which satisfies the timing specifications, is suitable.

## Erasure Characteristics

Erasure of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent erasure. **For this and secondary effects, it is recommended that an opaque label be placed over the window.** For elevated temperature or environments where solvents are being used, apply Kapton tape Fluorglas part number 2345-5, or equivalent.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 angstroms) to an integrated dose of at least 15W-s/cm<sup>2</sup>. Exposing the EPROM to an ultraviolet lamp of 12,000µW/cm<sup>2</sup> rating for 20 to 39 minutes, at a distance of about 1 inch, should be sufficient.

Erasure leaves the array in an all 1s state.

Table 2. EPROM Programming Modes<sup>1,2,3</sup>

MODE	RST	PSEN	ALE/PROG	$\overline{EA}/V_{PP}$	P2.7	P2.6	P3.7	P3.6
Read signature	1	0	1	1	0	0	0	0
Program code data	1	0	0 <sup>4</sup>	$V_{PP}$	1	0	1	1
Verify code data	1	0	1	1	0	1	1	1
Pgm encryption table	1	0	0 <sup>4</sup>	$V_{PP}$	1	0	1	0
Pgm security bit 1	1	0	0 <sup>4</sup>	$V_{PP}$	1	1	1	1
Pgm security bit 2	1	0	0 <sup>4</sup>	$V_{PP}$	1	1	0	0

## NOTES:

- '0' = Valid low for that pin, '1' = valid high for that pin.
- $V_{PP}$  = 12.75V  $\pm$  0.25V.
- $V_{CC}$  = 5V  $\pm$  10% during programming and verification.
- ALE/PROG receives 5 to 25 programming pulses while  $V_{PP}$  is held at 12.75V. Each programming pulse is low for 100µs ( $\pm$  10µs) and high for a minimum of 10µs.

™Trademark phrase of Intel Corporation.

## CMOS single-chip 3.0V 8-bit microcontroller 83L51FB/87L51FB

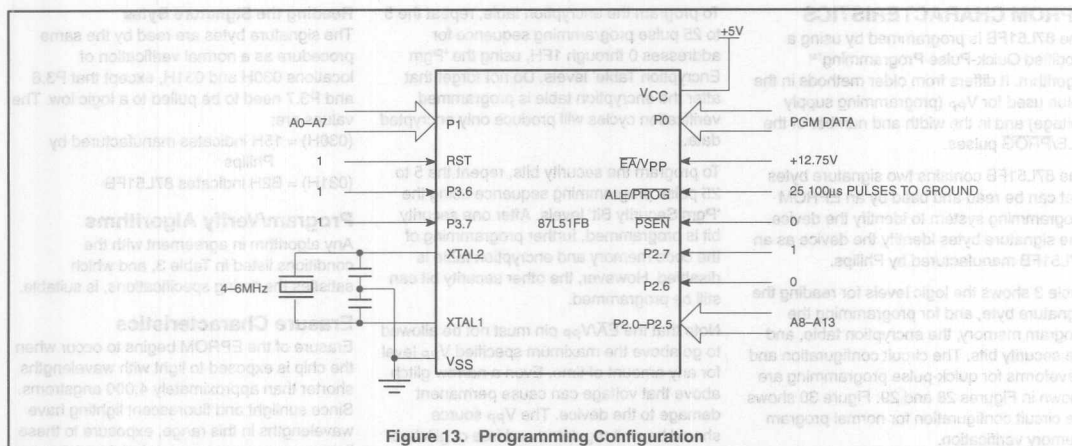


Figure 13. Programming Configuration

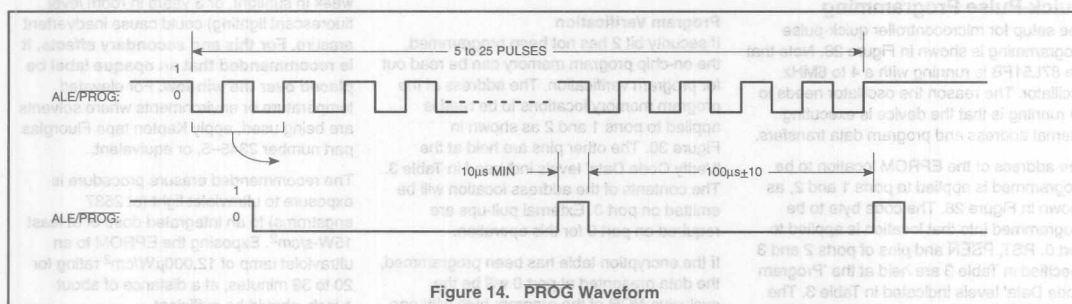


Figure 14. PROG Waveform

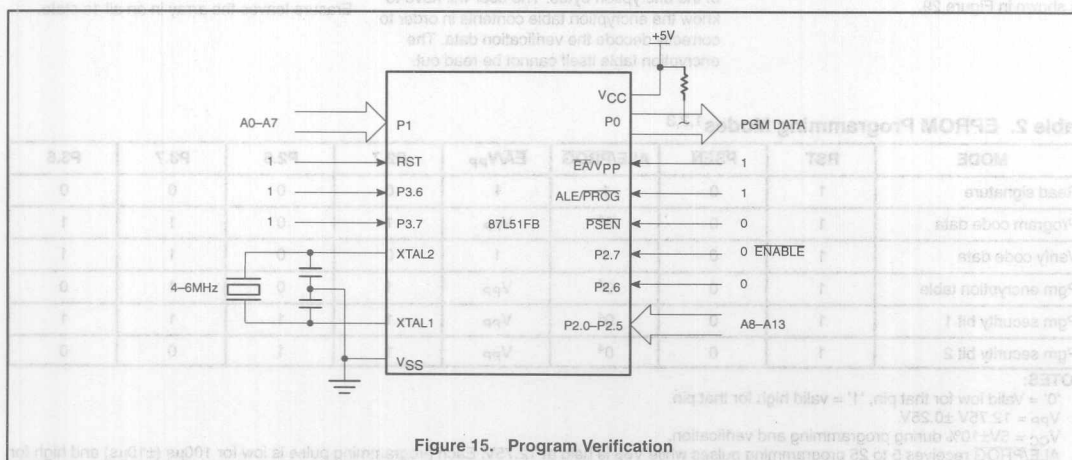


Figure 15. Program Verification



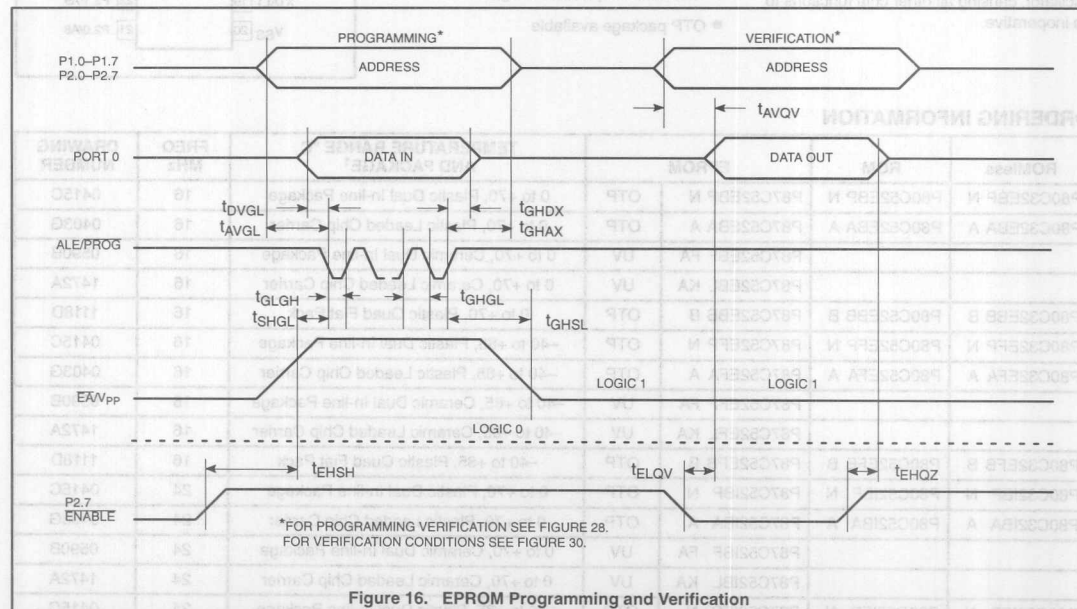
## CMOS single-chip 3.0V 8-bit microcontroller

83L51FB/87L51FB

## EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

 $T_{amb} = 21^{\circ}\text{C}$  to  $+27^{\circ}\text{C}$ ,  $V_{CC} = 5V \pm 10\%$ ,  $V_{SS} = 0V$  (See Figure 31)

SYMBOL	PARAMETER	MIN	MAX	UNIT
$V_{PP}$	Programming supply voltage	12.5	13.0	V
$I_{PP}$	Programming supply current		50	mA
$1/t_{CLCL}$	Oscillator frequency	4	6	MHz
$t_{AVGL}$	Address setup to PROG low	$48t_{CLCL}$		
$t_{GHAX}$	Address hold after PROG	$48t_{CLCL}$		
$t_{DVGL}$	Data setup to PROG low	$48t_{CLCL}$		
$t_{GHDx}$	Data hold after PROG	$48t_{CLCL}$		
$t_{EHS}$	P2.7 (ENABLE) high to $V_{PP}$	$48t_{CLCL}$		
$t_{SHGL}$	$V_{PP}$ setup to PROG low	10		$\mu\text{s}$
$t_{GHS}$	$V_{PP}$ hold after PROG	10		$\mu\text{s}$
$t_{GLGH}$	PROG width	90	110	$\mu\text{s}$
$t_{AVQV}$	Address to data valid		$48t_{CLCL}$	
$t_{ELOZ}$	ENABLE low to data valid		$48t_{CLCL}$	
$t_{EHQZ}$	Data float after ENABLE	0	$48t_{CLCL}$	
$t_{GHGL}$	PROG high to PROG low	10		$\mu\text{s}$





# CMOS single-chip 8-bit microcontrollers 80C32/80C52/87C52

## DESCRIPTION

The Philips 80C32/80C52/87C52 is a high-performance microcontroller fabricated with Philips high-density CMOS technology. The CMOS 8XC52 is functionally compatible with the NMOS SCN- 8032/8052 microcontrollers. The Philips CMOS technology combines the high speed and density characteristics of HMOS with the low power attributes of CMOS. Philips epitaxial substrate minimizes latch-up sensitivity.

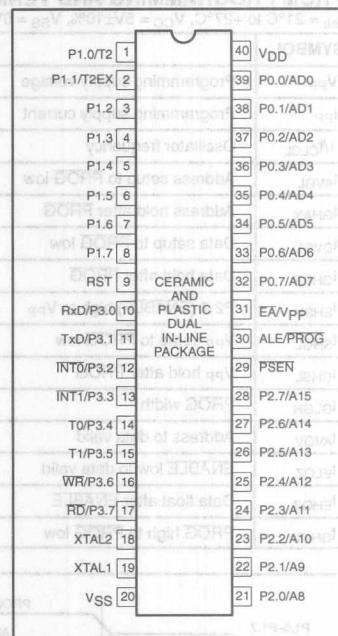
The 8XC52 contains an 8k × 8 ROM (80C52) EPROM (87C52), a 256 × 8 RAM, 32 I/O lines, three 16-bit counter/timers, a six-source, two-priority level nested interrupt structure, a serial I/O port for either multi-processor communications, I/O expansion or full duplex UART, and on-chip oscillator and clock circuits.

In addition, the 8XC52 has two software selectable modes of power reduction—idle mode and power-down mode. The idle mode freezes the CPU while allowing the RAM, timers, serial port, and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.

## FEATURES

- 80C51 based architecture
- 8032/8052 compatible
  - 8k × 8 ROM (80C52)
  - 8k × 8 EPROM (87C52)
  - ROMless (80C32)
  - 256 × 8 RAM
  - Three 16-bit counter/timers
  - Full duplex serial channel
  - Boolean processor
- Memory addressing capability
  - 64k ROM and 64k RAM
- Power control modes:
  - Idle mode
  - Power-down mode
- CMOS and TTL compatible
- Two speed ranges:
  - 3.5 to 16MHz
  - 3.5 to 24MHz
- Five package styles
- Extended temperature ranges
- OTP package available

## PIN CONFIGURATIONS



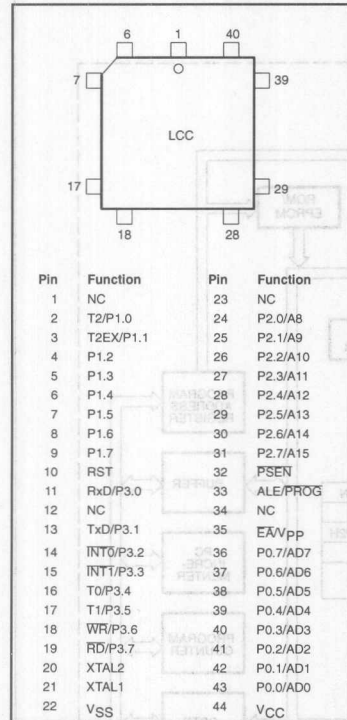
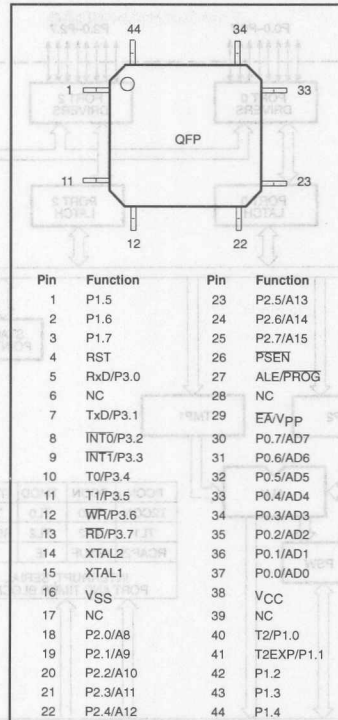
## ORDERING INFORMATION

ROMless	ROM	EPROM		TEMPERATURE RANGE °C AND PACKAGE <sup>1</sup>	FREQ MHz	DRAWING NUMBER
P80C32EBP N	P80C52EBP N	P87C52EBP N	OTP	0 to +70, Plastic Dual In-line Package	16	0415C
P80C32EBA A	P80C52EBA A	P87C52EBA A	OTP	0 to +70, Plastic Leaded Chip Carrier	16	0403G
		P87C52EBF FA	UV	0 to +70, Ceramic Dual In-line Package	16	0590B
		P87C52EBL KA	UV	0 to +70, Ceramic Leaded Chip Carrier	16	1472A
P80C32EBB B	P80C52EBB B	P87C52EBB B	OTP	0 to +70, Plastic Quad Flat Pack	16	1118D
P80C32EFP N	P80C52EFP N	P87C52EFP N	OTP	–40 to +85, Plastic Dual In-line Package	16	0415C
P80C32EFA A	P80C52EFA A	P87C52EFA A	OTP	–40 to +85, Plastic Leaded Chip Carrier	16	0403G
		P87C52EFF FA	UV	–40 to +85, Ceramic Dual In-line Package	16	0590B
		P87C52EFL KA	UV	–40 to +85, Ceramic Leaded Chip Carrier	16	1472A
P80C32EFB B	P80C52EFB B	P87C52EFB B	OTP	–40 to +85, Plastic Quad Flat Pack	16	1118D
P80C32IBP N	P80C52IBP N	P87C52IBP N	OTP	0 to +70, Plastic Dual In-line Package	24	0415C
P80C32IBA A	P80C52IBA A	P87C52IBA A	OTP	0 to +70, Plastic Leaded Chip Carrier	24	0403G
		P87C52IBF FA	UV	0 to +70, Ceramic Dual In-line Package	24	0590B
		P87C52IBL KA	UV	0 to +70, Ceramic Leaded Chip Carrier	24	1472A
P80C32IFP N	P80C52IFP N	P87C52IFP N	OTP	–40 to +85, Plastic Dual In-line Package	24	0415C
P80C32IFA A	P80C52IFA A	P87C52IFA A	OTP	–40 to +85, Plastic Leaded Chip Carrier	24	0403G
		P87C52IFF FA	UV	–40 to +85, Ceramic Dual In-line Package	24	0590B
		P87C52IFL KA	UV	–40 to +85, Ceramic Leaded Chip Carrier	24	1472A

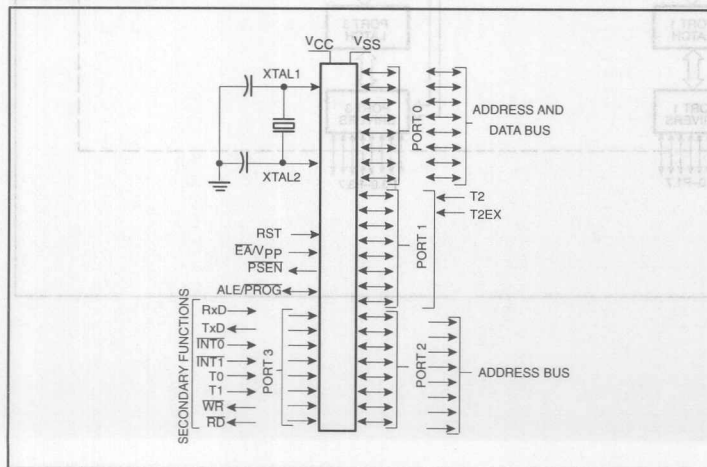
### NOTE:

1. OTP = One Time Programmable EPROM. UV = UV erasable EPROM

## CMOS single-chip 8-bit microcontrollers 80C32/80C52/87C52

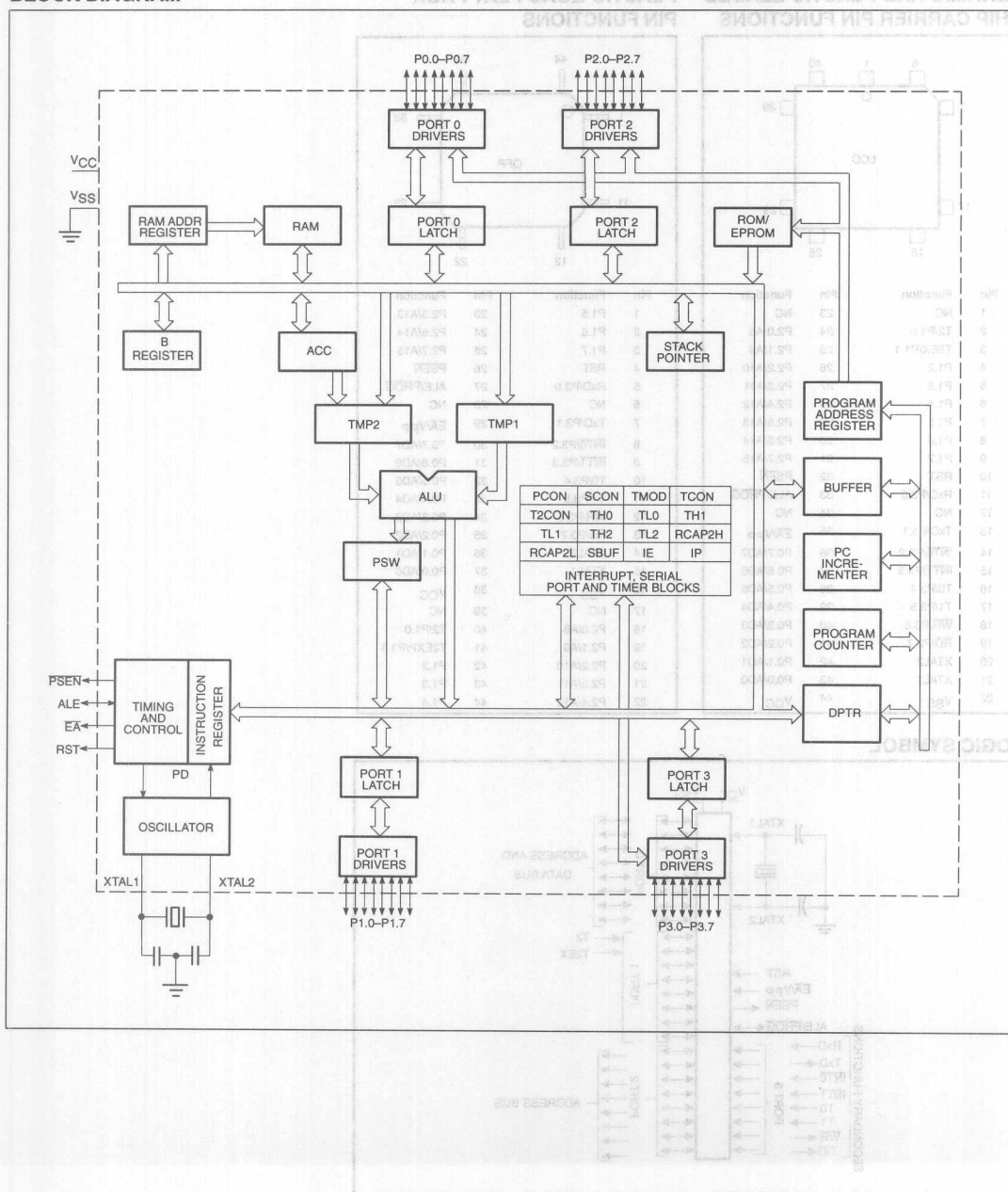
CERAMIC AND PLASTIC LEADED  
CHIP CARRIER PIN FUNCTIONSPLASTIC QUAD FLAT PACK  
PIN FUNCTIONS

## LOGIC SYMBOL



## CMOS single-chip 8-bit microcontrollers 80C32/80C52/87C52

## BLOCK DIAGRAM



## CMOS single-chip 8-bit microcontrollers 80C32/80C52/87C52

Table 1. 8XC52 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB							LSB	
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
DPTR:	Data pointer (2 bytes)										
DPH	Data pointer high	83H									00H
DPL	Data pointer low	82H									00H
			AF	AE	AD	AC	AB	AA	A9	A8	
IE*	Interrupt enable	A8H	EA	—	ET2	ES	ET1	EX1	ET0	EX0	0x000000B
			BF	BE	BD	BC	BB	BA	B9	B8	
IP*	Interrupt priority	B8H	—	—	PT2	PS	PT1	PX1	PT0	PX0	xx000000B
			87	86	85	84	83	82	81	80	
P0*	Port 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
			97	96	95	94	93	92	91	90	
P1*	Port 1	90H	—	—	—	—	—	—	T2EX	T2	FFH
			A7	A6	A5	A4	A3	A2	A1	A0	
P2*	Port 2	A0H	A15	A14	A13	A12	A11	A10	A9	A8	FFH
			B7	B6	B5	B4	B3	B2	B1	B0	
P3*	Port 3	B0H	RD	WR	T1	T0	INT1	INT0	TxD	RxD	FFH
PCON <sup>1</sup>	Power control	87H	SMOD	—	—	—	GF1	GF0	PD	IDL	0xxxxxxxB
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program status word	D0H	CY	AC	F0	RS1	RS0	OV	—	P	00H
RCAP2H#	Capture high	CBH									00H
RCAPL#	Capture low	CAH									00H
SBUF	Serial data buffer	99H									xxxxxxxB
			9F	9E	9D	9C	9B	9A	99	98	
SCON*	Serial controller	98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00H
SP	Stack pointer	81H									07H
			8F	8E	8D	8C	8B	8A	89	88	
TCON*	Timer control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
			CF	CE	CD	CC	CB	CA	C9	C8	
T2CON*#	Timer 2 control	C8H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	00H
TH0	Timer high 0	8CH									00H
TH1	Timer high 1	8DH									00H
TH2#	Timer high 2	CDH									00H
TL0	Timer low 0	8AH									00H
TL1	Timer low 1	8BH									00H
TL2#	Timer low 2	CCH									00H
TMOD	Timer mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H

\* Bit addressable

# SFRs are modified from or added to the 80C51 SFRs.

1. Bits GF1, GF0, PD, and IDL of the PCON register are not implemented in the NMOS 8XC52.

## CMOS single-chip 8-bit microcontrollers 80C32/80C52/87C52

## PIN DESCRIPTION

MNEMONIC	DIP	LCC	QFP	TYPE	NAME AND FUNCTION
V <sub>SS</sub>	20	22	16	I	<b>Ground:</b> 0V reference.
V <sub>CC</sub>	40	44	38	I	<b>Power Supply:</b> This is the power supply voltage for normal, idle, and power-down operation.
P0.0–0.7	39–32	43–36	37–30	I/O	<b>Port 0:</b> Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s. Port 0 also outputs the code bytes during program verification in the 87C52. External pull-ups are required during program verification.
P1.0–P1.7	1–8	2–9	40–44 1–3	I/O	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Pins P1.0 and P1.1 also. Port 1 also receives the low-order address byte during program memory verification. Port 1 also serves alternate functions for timer 2: <b>T2 (P1.0):</b> Timer/counter 2 external count input. <b>T2EX (P1.1):</b> Timer/counter 2 trigger input.
P2.0–P2.7	21–28	24–31	18–25	I/O	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register.
P3.0–P3.7	10–17	11, 13–19	5, 7–13	I/O	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 3 also serves the special features of the 80C51 family, as listed below: <b>RxD (P3.0):</b> Serial input port <b>TxD (P3.1):</b> Serial output port <b>INT0 (P3.2):</b> External interrupt <b>INT1 (P3.3):</b> External interrupt <b>T0 (P3.4):</b> Timer 0 external input <b>T1 (P3.5):</b> Timer 1 external input <b>WR (P3.6):</b> External data memory write strobe <b>RD (P3.7):</b> External data memory read strobe
RST	9	10	4	I	<b>Reset:</b> A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to V <sub>SS</sub> permits a power-on reset using only an external capacitor to V <sub>CC</sub> .
ALE/PROG	30	33	27	I/O	<b>Address Latch Enable/Program Pulse:</b> Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory. This pin is also the program pulse input (PROG) during EPROM programming.
PSEN	29	32	26	O	<b>Program Store Enable:</b> The read strobe to external program memory. When the device is executing code from the external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory. PSEN is not activated during fetches from internal program memory.
E <sub>A</sub> /V <sub>PP</sub>	31	35	29	I	<b>External Access Enable/Programming Supply Voltage:</b> E <sub>A</sub> must be externally held low to enable the device to fetch code from external program memory locations 0000H to 1FFFH. If E <sub>A</sub> is held high, the device executes from internal program memory unless the program counter contains an address greater than 1FFFH. This pin also receives the 12.75V programming supply voltage (V <sub>PP</sub> ) during EPROM programming.
XTAL1	19	21	15	I	<b>Crystal 1:</b> Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	20	14	O	<b>Crystal 2:</b> Output from the inverting oscillator amplifier.



## CMOS single-chip 8-bit microcontrollers 80C32/80C52/87C52

## DIFFERENCES FROM THE 80C51

## Special Function Registers

The special function register space is the same as the 80C51 except that the 80C52 contains the additional special function registers T2CON, RCAP2L, RCAP2H, TL2, and TH2. Since the standard 80C51 on-chip functions are identical in the 80C52, the SFR locations, bit locations, and operation are likewise identical. The only exceptions are in the interrupt mode and interrupt priority SFRs (see Table 1).

## Timer/Counters

In addition to timer/counters 0 and 1 of the 80C51, the 80C52 contains timer/counter 2. Like timers 0 and 1, timer 2 can operate as either an event timer or as an event counter. This is selected by bit C/T2 in the special function register T2CON (see Figure 1). It has three operating modes: capture, auto-load, and baud rate generator, which are selected by bits in the T2CON as shown in Table 2.

In the Capture Mode there are two options which are selected by bit EXEN2 in T2CON. If EXEN2 = 0, then Timer 2 is a 16-bit timer or counter which upon overflowing sets bit TF2, the Timer 2 overflow bit, which can be used to generate an interrupt. If EXEN2 = 1, then Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input T2EX causes the current value

in the Timer 2 registers, TL2 and TH2, to be captured into registers RCAP2L and RCAP2H, respectively. (RCAP2L and RCAP2H are new special function registers in the 80C52.) In addition, the transition at T2EX causes bit EXF2 in T2CON to be set, and EXF2 like TF2 can generate an interrupt. The Capture Mode is illustrated in Figure 2.

In the auto-reload mode, there are again two options, which are selected by bit EXEN2 in T2CON. If EXEN2 = 0, then when Timer 2 rolls over it not only sets TF2 but also causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2L and RCAP2H, which are preset by software. If EXEN2 = 1, then Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input T2EX will also trigger the 16-bit reload and set EXF2. The auto-reload mode is illustrated in Figure 3.

The baud rate generation mode is selected by RCLK = 1 and/or TCLK = 1. It will be described in conjunction with the serial port.

## Serial Port

The serial port of the 80C52 is identical to that of the 80C51 except that counter/timer 2 can be used to generate baud rates.

In the 80C52, Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (see Figure 1). Note that the baud rate for transmit and receive can be simultaneously different. Setting RCLK and/or

TCLK puts Timer into its baud rate generator mode, as shown in Figure 4.

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by software.

Now, the baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate as follows:

$$\text{Modes 1, 3 Baud Rate} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The timer can be configured for either "timer" or "counter" operation. In the most typical applications, it is configured for "timer" operation (C/T2 = 0). "Timer" operation is a little different for Timer 2 when it's being used as a baud rate generator. Normally, as a timer it would increment every machine cycle (thus at 1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time (thus at 1/2 the oscillator frequency). In that case the baud rate is given by the formula:

$$\text{Modes 1, 3 Baud Rate} = \frac{\text{Oscillator Frequency}}{32 \times (65536 - (\text{RCAP2H}, \text{RCAP2L}))}$$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

(MSB)				(LSB)			
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2

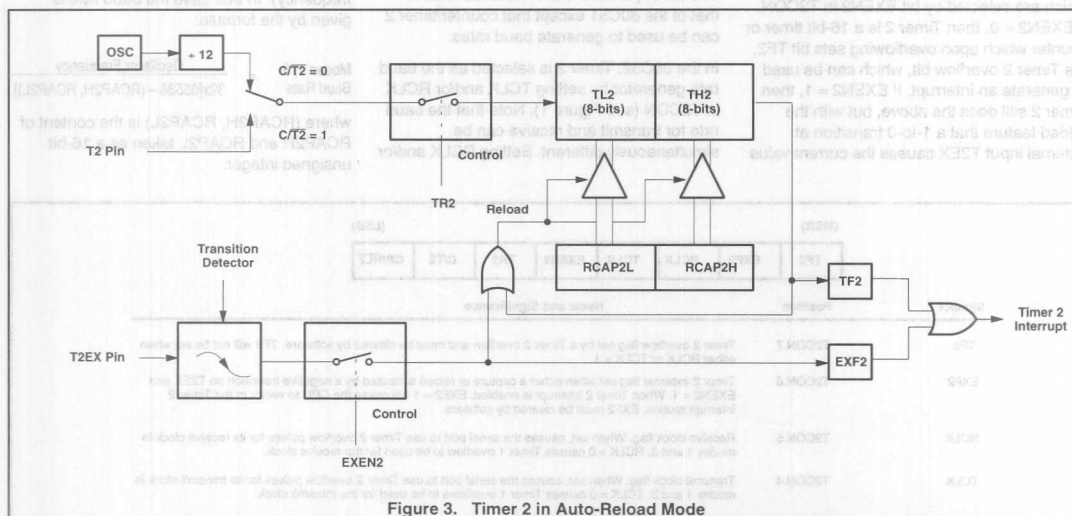
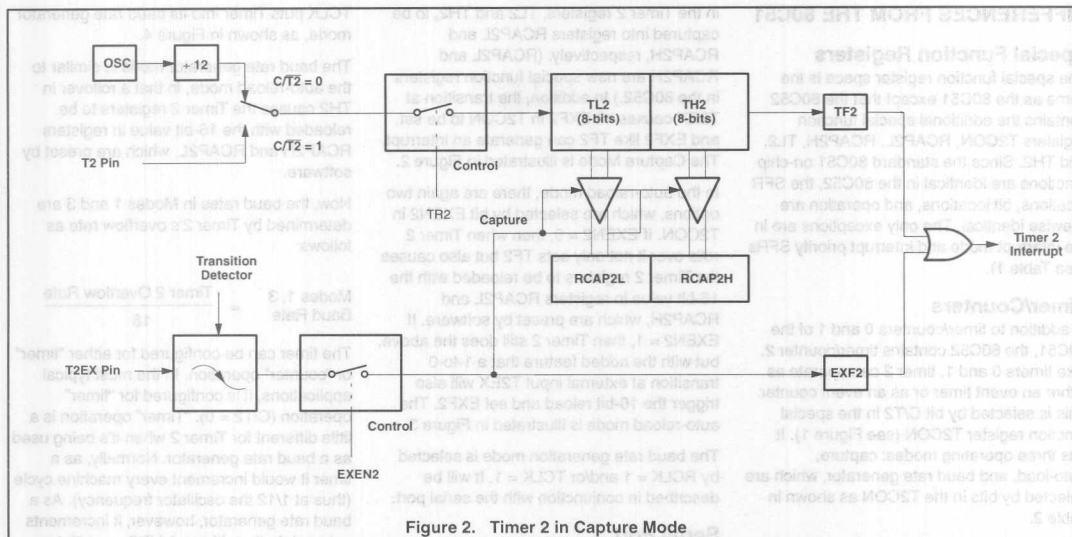
Symbol	Position	Name and Significance
TF2	T2CON.7	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK or TCLK = 1.
EXF2	T2CON.6	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software.
RCLK	T2CON.5	Receive clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.
TCLK	T2CON.4	Transmit clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	T2CON.3	Timer 2 external enable flag. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	T2CON.2	Start/stop control for Timer 2. A logic 1 starts the timer.
C/T2	T2CON.1	Timer or counter select. (Timer 2) 0 = Internal timer (OSC/12) 1 = External event counter (falling edge triggered).
CP/RL2	T2CON.0	Capture/Reload flag. When set, captures will occur on negative transitions at T2EX if EXEN2 = 1. When cleared, auto-reloads will occur either with Timer 2 overflows or negative transitions at T2EX when EXEN2 = 1. When either RCLK = 1 or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

**Figure 1. Timer/Counter 2 (T2CON) Control Register**

Figure 1. Timer/Counter 2 (T2CON) Control Register





## CMOS single-chip 8-bit microcontrollers 80C32/80C52/87C52

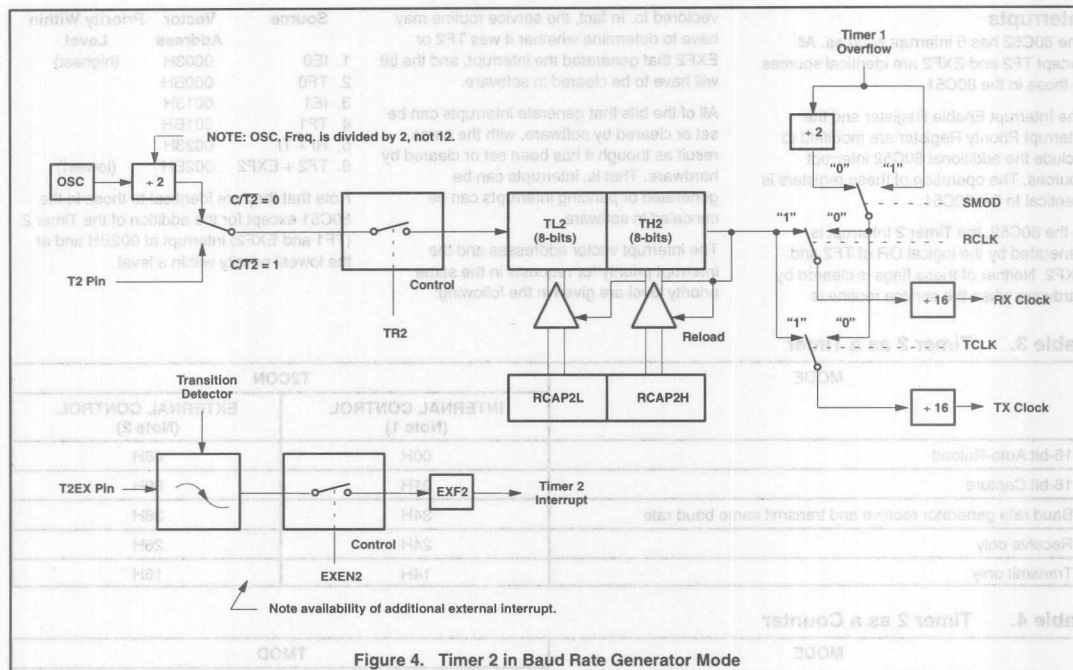


Table 2. Timer 2 Operating Modes

RCLK + RCLK	CP/RL2	TR2	MODE
0	0	1	16-bit Auto-reload
0	1	1	16-bit Capture
1	X	1	Baud rate generator
X	X	0	(off)

Timer 2 as a baud rate generator is shown in Figure 4. This figure is valid only if RCLK + TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2, and will not generate an interrupt. Therefore, the Timer 2 interrupt does not have to be disabled when Timer 2 is in the baud rate generator mode. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt, if desired.

It should be noted that when Timer 2 is running (TR2 = 1) in "timer" function in the baud rate generator mode, one should not try to read or write TH2 or TL2. Under these conditions the timer is being incremented

every state time, and the results of a read or write may not be accurate. The RCAP registers may be read, but should not be written to, because a write might overlap a reload and cause write and/or reload errors. Turn the timer off (clear TR2) before accessing the Timer 2 or RCAP registers, in this case.

#### Timer/Counter 2 Set-up

Except for the baud rate generator mode, the values given for T2CON do not include the setting of the TR2 bit. Therefore, bit TR2 must be set, separately, to turn the timer on. See Table 3 for set-up of timer 2 as a timer. See Table 4 for set-up of timer 2 as a counter.

#### Using Timer/Counter 2 to Generate Baud Rates

For this purpose, Timer 2 must be used in the baud rate generating mode. If Timer 2 is being clocked through pin T2 (P1.0) the baud rate is:

$$\text{Baud Rate} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

And if it is being clocked internally, the baud rate is:

$$\text{Baud Rate} = \frac{\text{Oscillator Frequency}}{32 \times (65536 - (\text{RCAP2H} \times 256 + \text{RCAP2L}))}$$

To obtain the reload value for RCAP2H and RCAP2L, the above equation can be rewritten as:

$$\begin{aligned} \text{RCAP2H} &= 65536 - \frac{\text{Oscillator Frequency}}{32 \times \text{Baud Rate}} \\ \text{RCAP2L} &= \end{aligned}$$

CMOS single-chip 8-bit microcontrollers 80C32/80C52/87C52

Interrupts

The 80C52 has 6 interrupt sources. All except TF2 and EXF2 are identical sources to those in the 80C51.

The Interrupt Enable Register and the Interrupt Priority Register are modified to include the additional 80C52 interrupt sources. The operation of these registers is identical to the 80C51.

In the 80C52, the Timer 2 Interrupt is generated by the logical OR of TF2 and EXF2. Neither of these flags is cleared by hardware when the service routine is

vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and the bit will have to be cleared in software.

All of the bits that generate interrupts can be set or cleared by software, with the same result as though it has been set or cleared by hardware. That is, interrupts can be generated or pending interrupts can be canceled in software.

The interrupt vector addresses and the interrupt priority for requests in the same priority level are given in the following:

Source	Vector Address	Priority Within Level
1. IE0	0003H	(highest)
2. TF0	000BH	
3. IE1	0013H	
4. TF1	001BH	
5. RI + TI	0023H	(lowest)
6. TF2 + EXF2	002BH	

Note that they are identical to those in the 80C51 except for the addition of the Timer 2 (TF1 and EXF2) interrupt at 002BH and at the lowest priority within a level.

Table 3. Timer 2 as a Timer

MODE	T2CON	
	INTERNAL CONTROL (Note 1)	EXTERNAL CONTROL (Note 2)
16-bit Auto-Reload	00H	08H
16-bit Capture	01H	09H
Baud rate generator receive and transmit same baud rate	34H	36H
Receive only	24H	26H
Transmit only	14H	16H

Table 4. Timer 2 as a Counter

MODE	TMOD	
	INTERNAL CONTROL (Note 1)	EXTERNAL CONTROL (Note 2)
16-bit	02H	0AH
Auto-Reload	03H	0BH

NOTES:

1. Capture/reload occurs only on timer/counter overflow.
2. Capture/reload occurs on timer/counter overflow and a 1-to-0 transition on T2EX (P1.1) pin except when timer 2 is used in the baud rate generator mode.

Using Timer/Counter 2 to Generate Baud Rates  
For this purpose, Timer 2 must be used in the baud rate generator mode. If Timer 2 is being clocked through pin T2 (P1.0), the baud rate is:  
$$\text{Baud Rate} = \frac{\text{Timer 2 Overflow Rate}}{16}$$
  
And if it is being clocked internally, the baud rate is:  
$$\text{Baud Rate} = \frac{\text{Oscillator Frequency}}{32 \times \text{Baud Rate}}$$
  
To obtain the reload value for RCAP2H and RCAP2L, the above equation can be rewritten as:

Timer/Counter 2 Set-up  
Except for the baud rate generator mode, the values given for T2CON do not include the reload of the T2H or T2L. Therefore, the values must be set separately to run the timer. See Table 4 for set-up of timer 2 as a counter.

Timer 2 as a Baud Rate Generator is shown in Figure 4. This figure is valid only if RCAP2H = 1 in T2CON, that is, a reload is required. This does not set T2H and will not generate an interrupt. Therefore, the Timer 2 interrupt does not have to be disabled when Timer 2 is in the baud rate generator mode. Note that the T2EX pin is set to a 1-to-0 transition in the baud rate generator mode. Note that T2EX will set EXF2 but will not cause a reload (RCAP2H, RCAP2L) if T2H = 1. Thus, when Timer 2 is in the baud rate generator mode, T2EX can be used as an external interrupt, if desired.

## CMOS single-chip 8-bit microcontrollers

## 80C32/80C52/87C52

**OSCILLATOR CHARACTERISTICS**

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator, as shown in the Logic Symbol, page 285.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

**RESET**

A reset is accomplished by holding the RST pin high for at least two machine cycles (24

oscillator periods), while the oscillator is running. To insure a good power-up reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few milliseconds) plus two machine cycles.

**IDLE MODE**

In idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

**POWER-DOWN MODE**

In the power-down mode, the oscillator is stopped and the instruction to invoke power-down is the last instruction executed. Only the contents of the on-chip RAM are preserved. A hardware reset is the only way to terminate the power-down mode. The control bits for the reduced power modes are in the special function register PCON.

**DESIGN CONSIDERATIONS**

At power-on, the voltage on  $V_{CC}$  and RST must come up at the same time for a proper start-up.

Table 5 shows the state of I/O ports during low current operating modes.

**ROM CODE SUBMISSION**

When submitting ROM code for the 80C52, the following must be specified:

1. 8k byte user ROM data
2. 32 byte ROM encryption key
3. ROM security bits.

ADDRESS	CONTENT	BIT(S)	COMMENT
0000H to 1FFFFH	DATA	7:0	User ROM Data
2000H to 201FH	KEY	7:0	ROM Encryption Key FFH = no encryption
2020H	SEC	0	ROM Security Bit 1
2020H	SEC	1	ROM Security Bit 2 0 = enable security 1 = disable security

**Security Bit 1:** When programmed, this bit has two effects on masked ROM parts:

1. External MOV<sub>C</sub> is disabled, and
2. EA# is latched on Reset.

**Security Bit 2:** When programmed, this bit inhibits Verify User ROM.

**Table 5. External Pin Status During Idle and Power-Down Modes**

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

CMOS single-chip 8-bit microcontrollers 80C32/80C52/87C52

Electrical Deviations from Commercial Specifications for Extended Temperature Range (87C52)  
DC and AC parameters not included here are the same as in the commercial temperature range table.

DC ELECTRICAL CHARACTERISTICS

T<sub>amb</sub> = -40°C to +85°C, V<sub>CC</sub> = 5V ±10%, V<sub>SS</sub> = 0V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			MIN	MAX	
V <sub>IL</sub>	Input low voltage, except EA		-0.5	0.2V <sub>CC</sub> -0.15	V
V <sub>IL1</sub>	Input low voltage to EA		0	0.2V <sub>CC</sub> -0.35	V
V <sub>IH</sub>	Input high voltage, except XTAL1, RST		0.2V <sub>CC</sub> +1	V <sub>CC</sub> +0.5	V
V <sub>IH1</sub>	Input high voltage to XTAL1, RST		0.7V <sub>CC</sub> +0.1	V <sub>CC</sub> +0.5	V
I <sub>IL</sub>	Logical 0 input current, ports 1, 2, 3	V <sub>IN</sub> = 0.45V		-75	μA
I <sub>TL</sub>	Logical 1-to-0 transition current, ports 1, 2, 3	V <sub>IN</sub> = 2.0V		-750	μA
I <sub>CC</sub>	Power supply current: Active mode Idle mode Power-down mode	V <sub>CC</sub> = 4.5-5.5V, Frequency range = 3.5 to 16MHz		19 6 50	mA mA μA

ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>

PARAMETER	RATING	UNIT
Operating temperature under bias	0 to +70 or -40 to +85	°C
Storage temperature range	-65 to +150	°C
Voltage on EA/V <sub>PP</sub> pin to V <sub>SS</sub>	0 to +13.0	V
Voltage on any other pin to V <sub>SS</sub>	-0.5 to +6.5	V
Maximum I <sub>OL</sub> per I/O pin	15	mA
Power dissipation (based on package heat transfer limitations, not device power consumption)	1.5	W

NOTES:

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
- This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
- Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V<sub>SS</sub> unless otherwise noted.

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Flash	External	1	1	Flash	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Flash	Data	Data	Data



## CMOS single-chip 8-bit microcontrollers 80C32/80C52/87C52

## DC ELECTRICAL CHARACTERISTICS

$T_{amb} = 0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$  or  $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$  (87C52)

$T_{amb} = 0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$  or  $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 20\%$ ,  $V_{SS} = 0\text{V}$  (80C32/80C52)

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP <sup>1</sup>	MAX	
$V_{IL}$	Input low voltage, except $\overline{EA}^7$		-0.5		$0.2V_{CC}-0.1$	V
$V_{IL1}$	Input low voltage to $\overline{EA}^7$		0		$0.2V_{CC}-0.3$	V
$V_{IH}$	Input high voltage, except XTAL1, RST <sup>7</sup>		$0.2V_{CC}+0.9$		$V_{CC}+0.5$	V
$V_{IH1}$	Input high voltage, XTAL1, RST <sup>7</sup>		$0.7V_{CC}$		$V_{CC}+0.5$	V
$V_{OL}$	Output low voltage, ports 1, 2, 3 <sup>9</sup>	$I_{OL} = 1.6\text{mA}^2$			0.45	V
$V_{OL1}$	Output low voltage, port 0, ALE, PSEN <sup>9</sup>	$I_{OL} = 3.2\text{mA}^2$			0.45	V
$V_{OH}$	Output high voltage, ports 1, 2, 3, ALE, PSEN <sup>3</sup>	$I_{OH} = -60\mu\text{A}$	2.4			V
		$I_{OH} = -25\mu\text{A}$	$0.75V_{CC}$			V
		$I_{OH} = -10\mu\text{A}$	$0.9V_{CC}$			V
$V_{OH1}$	Output high voltage (port 0 in external bus mode)	$I_{OH} = -800\mu\text{A}$	2.4			V
		$I_{OH} = -300\mu\text{A}$	$0.75V_{CC}$			V
		$I_{OH} = -80\mu\text{A}$	$0.9V_{CC}$			V
$I_{IL}$	Logical 0 input current, ports 1, 2, 3 <sup>7</sup>	$V_{IN} = 0.45\text{V}$			-50	$\mu\text{A}$
$I_{TL}$	Logical 1-to-0 transition current, ports 1, 2, 3 <sup>7</sup>	See note 4			-650	$\mu\text{A}$
$I_L$	Input leakage current, port 0	$V_{IN} = V_{IL}$ or $V_{IH}$			$\pm 10$	$\mu\text{A}$
$I_{CC}$	Power supply current: <sup>7</sup> Active mode @ 12MHz <sup>5</sup> Idle mode @ 12MHz Power-down mode	See note 6		11.5	19	mA
				1.3	4	mA
				3	50	$\mu\text{A}$
$R_{RST}$	Internal reset pull-down resistor		50		300	k $\Omega$
$C_{IO}$	Pin capacitance				10	pF

## NOTES:

- Typical ratings are not guaranteed. The values listed are at room temperature, 5V.
- Capacitive loading on ports 0 and 2 may cause spurious noise to be superimposed on the  $V_{OL}$ s of ALE and ports 1 and 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100pF), the noise pulse on the ALE pin may exceed 0.8V. In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.  $I_{OL}$  can exceed these conditions provided that no single output sinks more than 5mA and no more than two outputs exceed the test conditions.
- Capacitive loading on ports 0 and 2 may cause the  $V_{OH}$  on ALE and PSEN to momentarily fall below the  $0.9V_{CC}$  specification when the address bits are stabilizing.
- Pins of ports 1, 2 and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when  $V_{IN}$  is approximately 2V.
- $I_{CCMAX}$  at other frequencies is given by: Active mode:  $I_{CCMAX} = 1.43 \times \text{FREQ} + 1.9$ ; Idle mode:  $I_{CCMAX} = 0.14 \times \text{FREQ} + 2.31$ , where FREQ is the external oscillator frequency in MHz.  $I_{CCMAX}$  is given in mA. See Figure 12.
- See Figures 13 through 16 for  $I_{CC}$  test conditions.
- These values apply only to  $T_{amb} = 0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$ . For  $T_{amb} = -40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ , see table on previous page.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:  
Maximum  $I_{OL}$  per port pin: 15mA (\*NOTE: This is 85°C specification.)  
Maximum  $I_{OL}$  per 8-bit port: 26mA  
Maximum total  $I_{OL}$  for all outputs: 67mA  
If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.



## CMOS single-chip 8-bit microcontrollers 80C32/80C52/87C52

## AC ELECTRICAL CHARACTERISTICS

$T_{amb} = 0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$  or  $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$  (87C52)<sup>1, 2, 3</sup>

SYMBOL	FIGURE	PARAMETER	24MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
$f_{CLCL}$	5	Oscillator frequency: <b>Speed Versions</b> 8XC52 8XC52			3.5 3.5	16 24	MHz MHz
$t_{LHLL}$	5	ALE pulse width	43		$2t_{CLCL}-40$		ns
$t_{AVLL}$	5	Address valid to ALE low	28		$t_{CLCL}-13$		ns
$t_{LLAX}$	5	Address hold after ALE low	21		$t_{CLCL}-20$		ns
$t_{LLIV}$	5	ALE low to valid instruction in		101		$4t_{CLCL}-65$	ns
$t_{LLPL}$	5	ALE low to PSEN low	28		$t_{CLCL}-13$		ns
$t_{PLPH}$	5	PSEN pulse width	104		$3t_{CLCL}-20$		ns
$t_{PLIV}$	5	PSEN low to valid instruction in		79		$3t_{CLCL}-45$	ns
$t_{PXIX}$	5	Input instruction hold after PSEN	0		0		ns
$t_{PXIZ}$	5	Input instruction float after PSEN		31		$t_{CLCL}-10$	ns
$t_{AVIV}$	5	Address to valid instruction in		153		$5t_{CLCL}-55$	ns
$t_{PLAZ}$	5	PSEN low to address float		10		10	ns
<b>Data Memory</b>							
$t_{RLRH}$	6, 7	RD pulse width	149		$6t_{CLCL}-100$		ns
$t_{WLWH}$	6, 7	WR pulse width	149		$6t_{CLCL}-100$		ns
$t_{RLDV}$	6, 7	RD low to valid data in		118		$5t_{CLCL}-90$	ns
$t_{RHDZ}$	6, 7	Data hold after RD	0		0		ns
$t_{RHDZ}$	6, 7	Data float after RD		55		$2t_{CLCL}-28$	ns
$t_{LLDV}$	6, 7	ALE low to valid data in		183		$8t_{CLCL}-150$	ns
$t_{AVDV}$	6, 7	Address to valid data in		209		$9t_{CLCL}-165$	ns
$t_{LLWL}$	6, 7	ALE low to RD or WR low	74	174	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
$t_{AVWL}$	6, 7	Address valid to WR low or RD low	91		$4t_{CLCL}-75$		ns
$t_{QVWX}$	6, 7	Data valid to WR transition	21		$t_{CLCL}-20$		ns
$t_{WHQX}$	6, 7	Data hold after WR	21		$t_{CLCL}-20$		ns
$t_{RLAZ}$	6, 7	RD low to address float		0		0	ns
$t_{WHLH}$	6, 7	RD or WR high to ALE high	21	66	$t_{CLCL}-20$	$t_{CLCL}+25$	ns
<b>External Clock</b>							
$t_{CHCX}$	9	High time	20		20		ns
$t_{CLCX}$	9	Low time	20		20		ns
$t_{CLCH}$	9	Rise time		20		20	ns
$t_{CHCL}$	9	Fall time		20		20	ns
<b>Shift Register</b>							
$t_{XLXL}$	8	Serial port clock cycle time	499		$12t_{CLCL}$		ns
$t_{QVXH}$	8	Output data setup to clock rising edge	283		$10t_{CLCL}-133$		ns
$t_{XHDX}$	8	Output data hold after clock rising edge	3		$2t_{CLCL}-80$		ns
$t_{XHDX}$	8	Input data hold after clock rising edge	0		0		ns
$t_{XHDV}$	8	Clock rising edge to input data valid		283		$10t_{CLCL}-133$	ns

## NOTES:

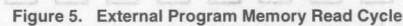
- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- Interfacing the 80C32/52 to devices with float times up to 45ns is permitted. This limited bus contention will not cause damage to Port 0 drivers.

## 80C32/80C52/87C52

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

P – PSEN  
Q – Output data  
R – RD signal  
t – Time  
V – Valid  
W – WR signal  
X – No longer a valid logic level  
Z – Float

**Examples:**  $t_{AVLL}$  = Time for address valid to ALE low.  
 $t_{LLPL}$  = Time for ALE low to  $\overline{PSEN}$  low.



## CMOS single-chip 8-bit microcontrollers 80C32/80C52/87C52

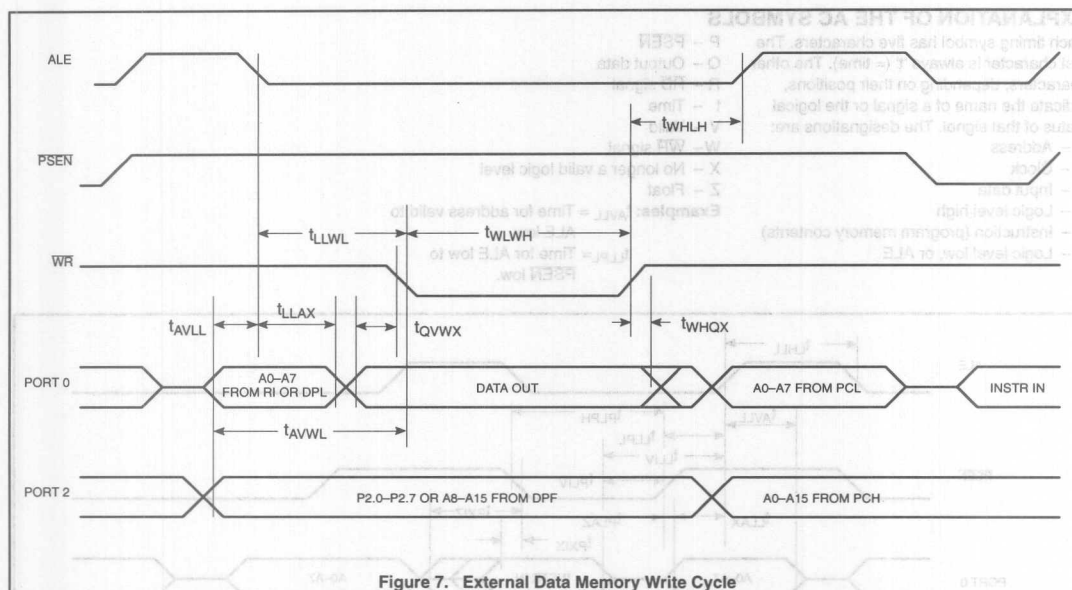


Figure 7. External Data Memory Write Cycle

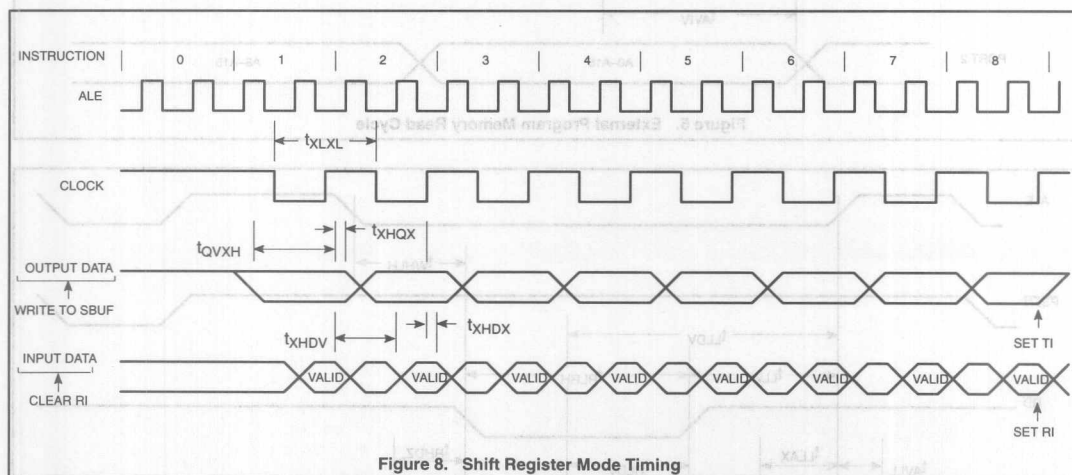
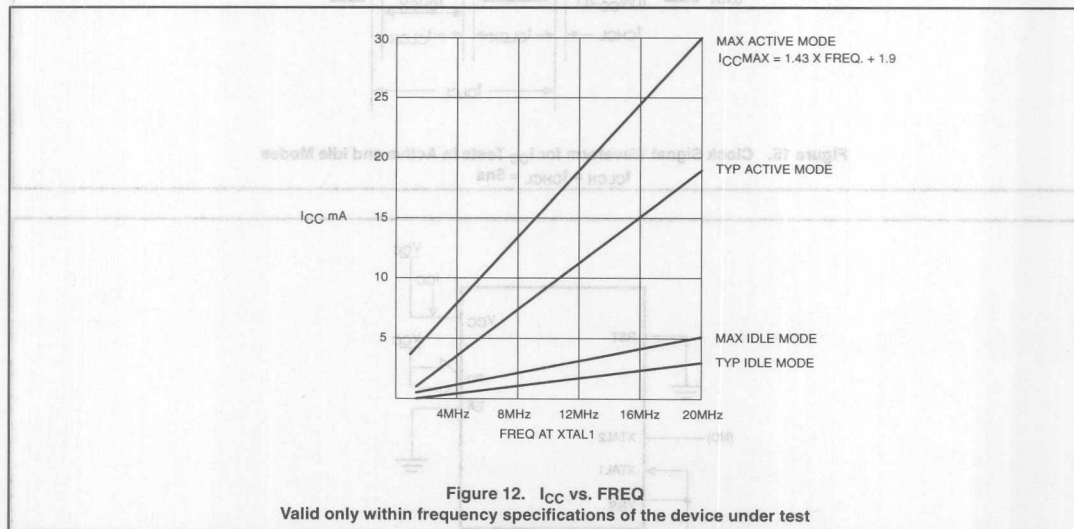
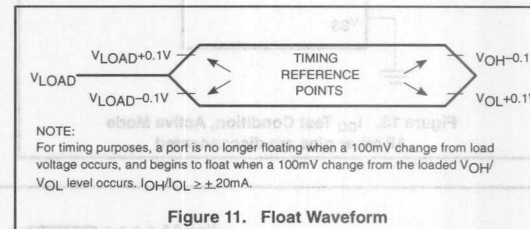
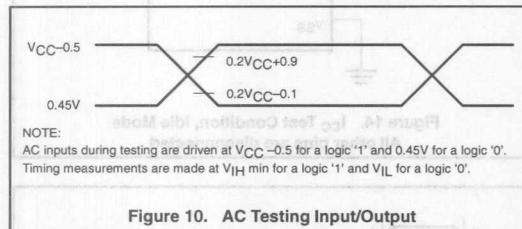
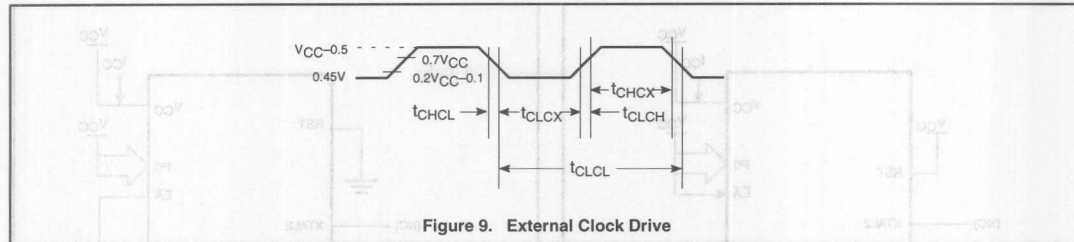
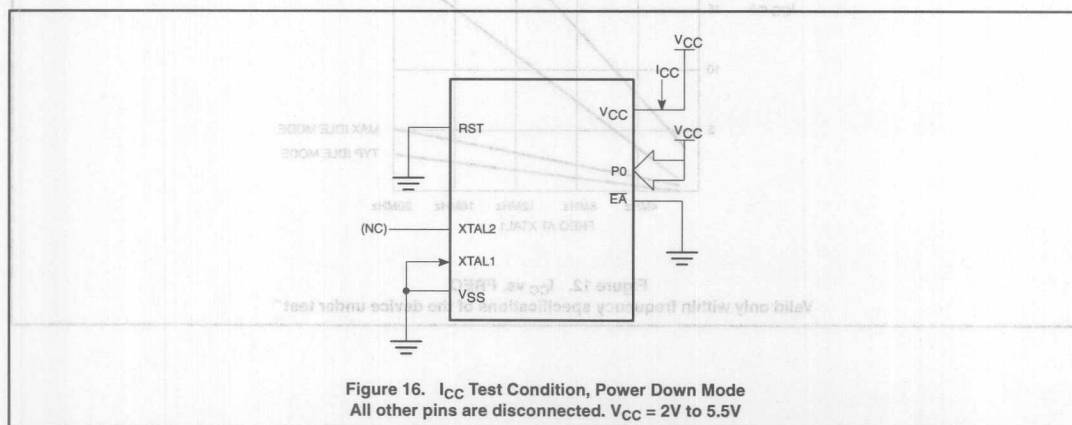
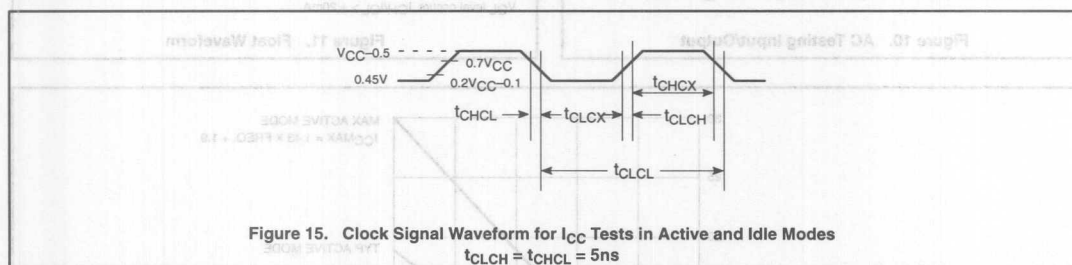
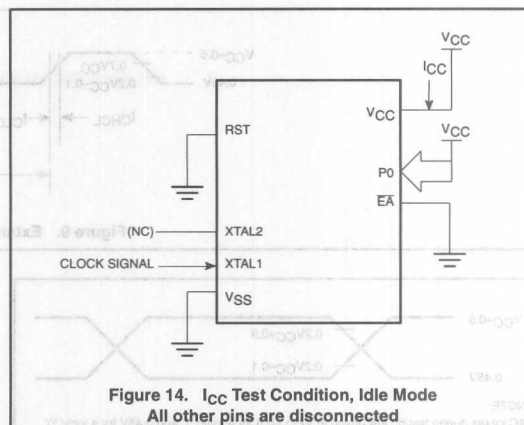
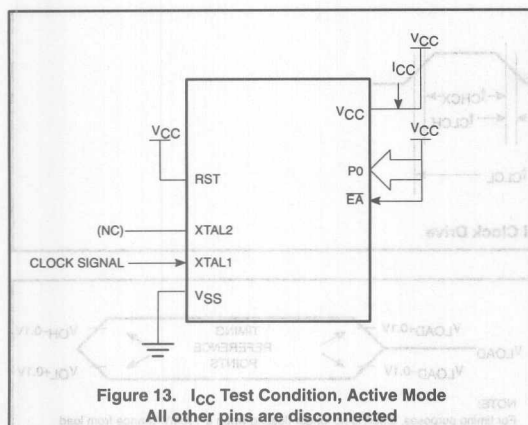


Figure 8. Shift Register Mode Timing

## CMOS single-chip 8-bit microcontrollers 80C32/80C52/87C52



## CMOS single-chip 8-bit microcontrollers 80C32/80C52/87C52



## CMOS single-chip 8-bit microcontrollers 80C32/80C52/87C52

**EPROM CHARACTERISTICS**

The 87C52 is programmed by using a modified Quick-Pulse Programming™ algorithm. It differs from older methods in the value used for  $V_{PP}$  (programming supply voltage) and in the width and number of the ALE/PROG pulses.

The 87C52 contains two signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes identify the device as an 87C52 manufactured by Philips.

Table 6 shows the logic levels for reading the signature byte, and for programming the program memory, the encryption table, and the security bits. The circuit configuration and waveforms for quick-pulse programming are shown in Figures 17 and 18. Figure 19 shows the circuit configuration for normal program memory verification.

**Quick-Pulse Programming**

The setup for microcontroller quick-pulse programming is shown in Figure 17. Note that the 87C52 is running with a 4 to 6MHz oscillator. The reason the oscillator needs to be running is that the device is executing internal address and program data transfers.

The address of the EPROM location to be programmed is applied to ports 1 and 2, as shown in Figure 17. The code byte to be programmed into that location is applied to port 0. RST, PSEN and pins of ports 2 and 3 specified in Table 6 are held at the 'Program Code Data' levels indicated in Table 6. The ALE/PROG is pulsed low 25 times as shown in Figure 18.

To program the encryption table, repeat the 25 pulse programming sequence for addresses 0 through 1FH, using the 'Pgm Encryption Table' levels. Do not forget that after the encryption table is programmed, verification cycles will produce only encrypted data.

To program the security bits, repeat the 25 pulse programming sequence using the 'Pgm Security Bit' levels. After one security bit is programmed, further programming of the code memory and encryption table is disabled. However, the other security bit can still be programmed.

Note that the  $\overline{EA}/V_{PP}$  pin must not be allowed to go above the maximum specified  $V_{PP}$  level for any amount of time. Even a narrow glitch above that voltage can cause permanent damage to the device. The  $V_{PP}$  source should be well regulated and free of glitches and overshoot.

**Program Verification**

If security bit 2 has not been programmed, the on-chip program memory can be read out for program verification. The address of the program memory locations to be read is applied to ports 1 and 2 as shown in Figure 19. The other pins are held at the 'Verify Code Data' levels indicated in Table 6. The contents of the address location will be emitted on port 0. External pull-ups are required on port 0 for this operation.

If the encryption table has been programmed, the data presented at port 0 will be the exclusive NOR of the program byte with one of the encryption bytes. The user will have to know the encryption table contents in order to correctly decode the verification data. The encryption table itself cannot be read out.

**Reading the Signature Bytes**

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. The values are:

(030H) = 15H indicates manufactured by Philips  
(031H) = 97H indicates 87C52

**Program/Verify Algorithms**

Any algorithm in agreement with the conditions listed in Table 6, and which satisfies the timing specifications, is suitable.

**Erase Characteristics**

Erase of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent erasure. **For this and secondary effects, it is recommended that an opaque label be placed over the window.** For elevated temperature or environments where solvents are being used, apply Kapton tape Fluorglas part number 2345-5, or equivalent.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 angstroms) to an integrated dose of at least 15W-s/cm<sup>2</sup>. Exposing the EPROM to an ultraviolet lamp of 12,000µW/cm<sup>2</sup> rating for 20 to 39 minutes, at a distance of about 1 inch, should be sufficient.

Erase leaves the array in an all 1s state.

**Table 6. EPROM Programming Modes**

MODE	RST	PSEN	ALE/PROG	$\overline{EA}/V_{PP}$	P2.7	P2.6	P3.7	P3.6
Read signature	1	0	1	1	0	0	0	0
Program code data	1	0	0*	$V_{PP}$	1	0	1	1
Verify code data	1	0	1	1	0	0	1	1
Pgm encryption table	1	0	0*	$V_{PP}$	1	0	1	0
Pgm security bit 1	1	0	0*	$V_{PP}$	1	1	1	1
Pgm security bit 2	1	0	0*	$V_{PP}$	1	1	0	0

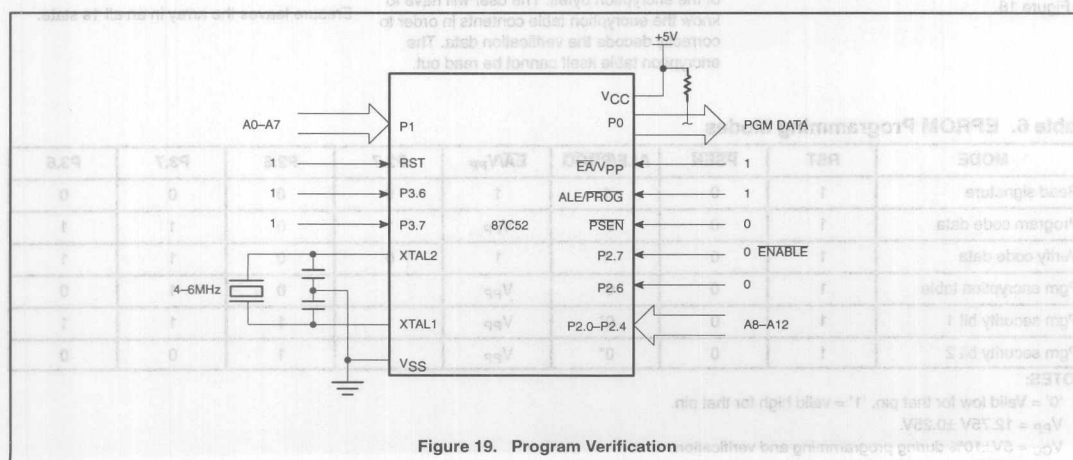
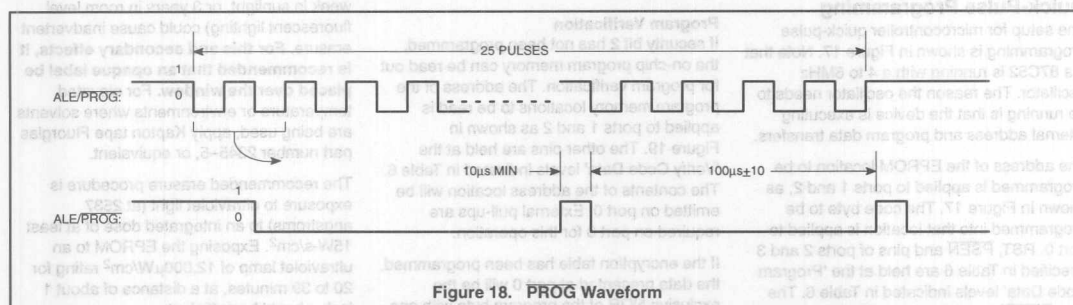
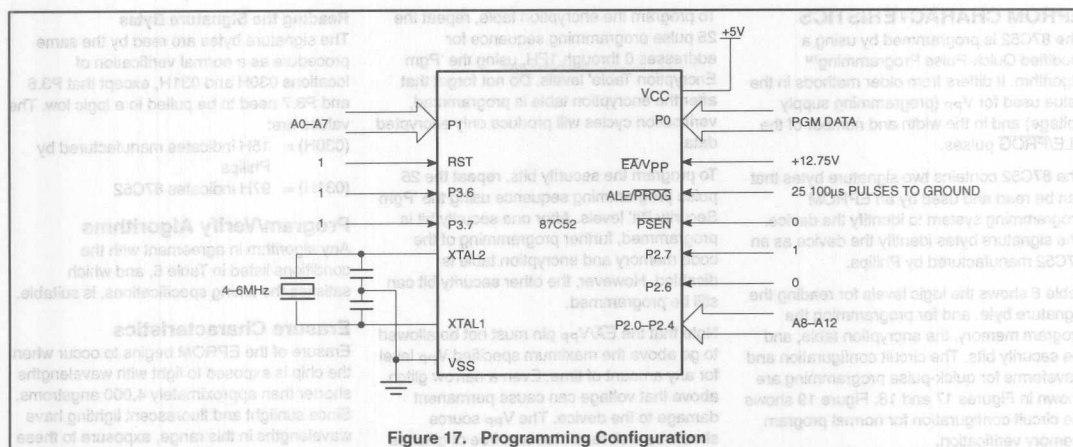
**NOTES:**

- '0' = Valid low for that pin, '1' = valid high for that pin.
- $V_{PP}$  = 12.75V  $\pm$  0.25V.
- $V_{CC}$  = 5V  $\pm$  10% during programming and verification.
- \*ALE/PROG receives 25 programming pulses while  $V_{PP}$  is held at 12.75V. Each programming pulse is low for 100µs ( $\pm$ 10µs) and high for a minimum of 10µs.

™Trademark phrase of Intel Corporation.



## CMOS single-chip 8-bit microcontrollers 80C32/80C52/87C52



## CMOS single-chip 8-bit microcontrollers 80C32/80C52/87C52

## EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

T<sub>amb</sub> = 21°C to +27°C, V<sub>CC</sub> = 5V±10%, V<sub>SS</sub> = 0V (See Figure 20)

SYMBOL	PARAMETER	MIN	MAX	UNIT
V <sub>PP</sub>	Programming supply voltage	12.5	13.0	V
I <sub>PP</sub>	Programming supply current		50	mA
1/t <sub>CLCL</sub>	Oscillator frequency	4	6	MHz
t <sub>AVGL</sub>	Address setup to PROG low	48t <sub>CLCL</sub>		
t <sub>GHAX</sub>	Address hold after PROG	48t <sub>CLCL</sub>		
t <sub>DVGL</sub>	Data setup to PROG low	48t <sub>CLCL</sub>		
t <sub>GHDX</sub>	Data hold after PROG	48t <sub>CLCL</sub>		
t <sub>ESH</sub>	P2.7 (ENABLE) high to V <sub>PP</sub>	48t <sub>CLCL</sub>		
t <sub>SHGL</sub>	V <sub>PP</sub> setup to PROG low	10		μs
t <sub>GHSL</sub>	V <sub>PP</sub> hold after PROG	10		μs
t <sub>GLGH</sub>	PROG width	90	110	μs
t <sub>AVQV</sub>	Address to data valid		48t <sub>CLCL</sub>	
t <sub>ELQZ</sub>	ENABLE low to data valid		48t <sub>CLCL</sub>	
t <sub>EHQZ</sub>	Data float after ENABLE	0	48t <sub>CLCL</sub>	
t <sub>GHGL</sub>	PROG high to PROG low	10		μs

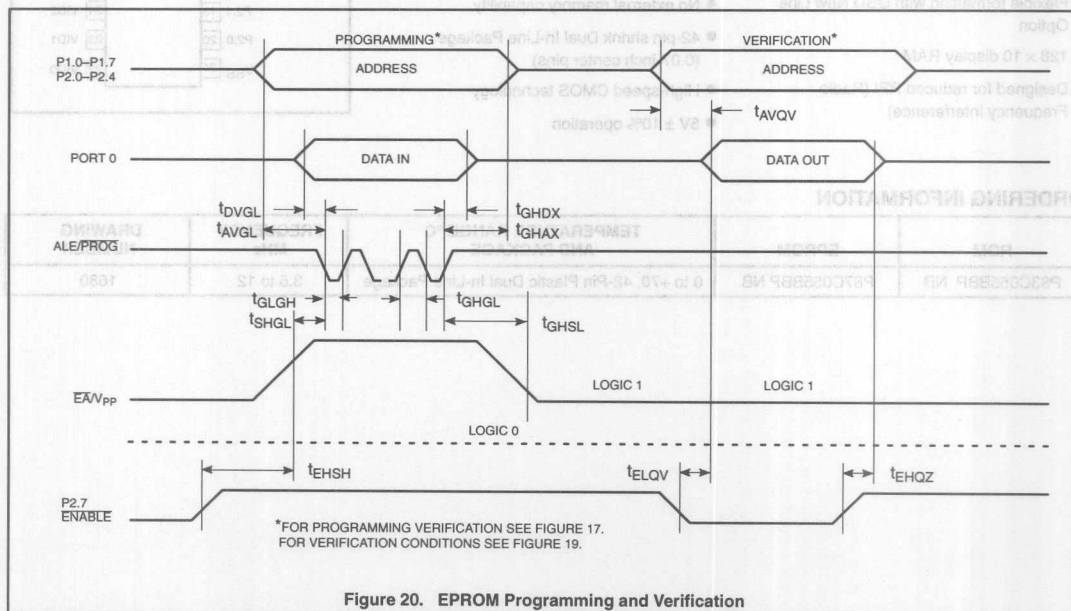


Figure 20. EPROM Programming and Verification

## DESCRIPTION

The Microcontroller for Television and Video (MTV) applications is a derivative of Philips' industry-standard 80C51 microcontroller that is intended for use as the central control mechanism in a television receiver or tuner. Providing tuner functions and an On Screen Display facility, it represents a next-generation replacement for the currently available parts.

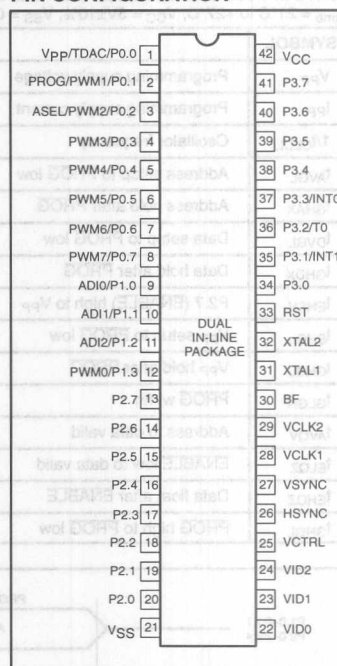
The MTV is available in either a 16K masked ROM, or a 16K One Time Programmable (OTP) EPROM version.

## FEATURES

- 16384 × 8 masked ROM (83C055), or 16384 × 8 OTP/EPROM (87C055)
- 256 × 8 RAM
- On Screen Display (OSD) Controller
- Three digital video outputs
- Multiplexer/mixer and background intensity controls
- Flexible formatting with OSD New Line Option
- 128 × 10 display RAM
- Designed for reduced RFI (Radio Frequency Interference)

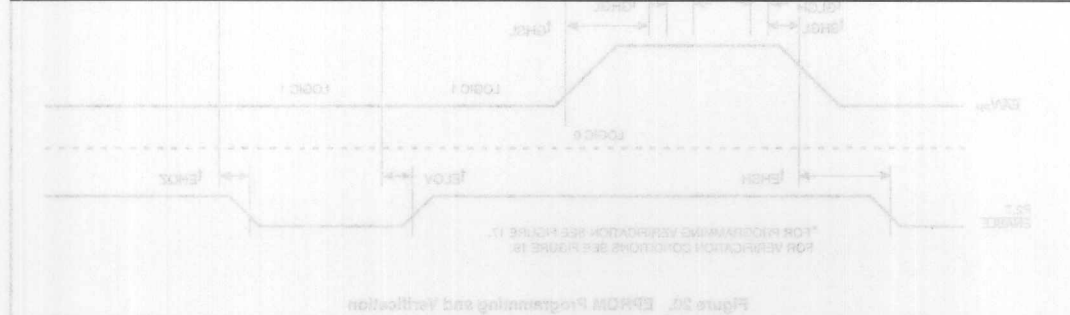
- 60 × 18 × 14 character generator ROM
- Eight text-shadowing modes
- Text color selectable per character
- Background color selectable per word
- Background color vs. video selectable per character
- Eight 6-bit pulse width modulators for analog voltage integration
- One 14-bit PWM for high-precision voltage integration
- D/A converter and comparator with three-input multiplexer
- Nine dedicated I/Os plus 28 port bits
- 15 port bits have alternate uses
- Four high-current open-drain port outputs
- 12 high-voltage (+12V) open drain outputs
- Programmable video input and output polarities
- 80C51 instruction set
- No external memory capability
- 42-pin shrink Dual In-Line Package (0.07-inch center pins)
- High-speed CMOS technology
- 5V ± 10% operation

## PIN CONFIGURATION



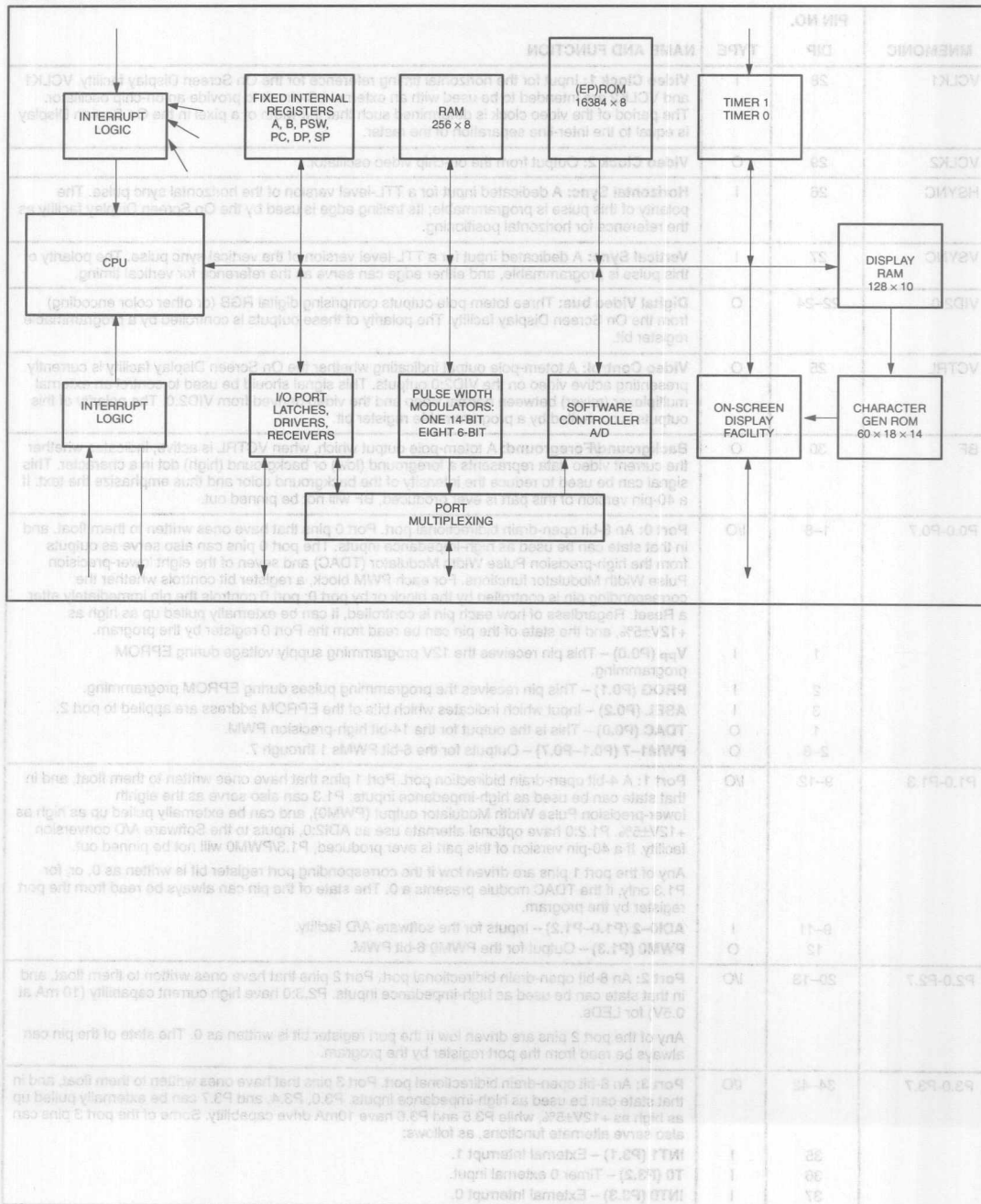
## ORDERING INFORMATION

ROM	EPROM	TEMPERATURE RANGE °C AND PACKAGE	FREQUENCY MHz	DRAWING NUMBER
P83C055BBP NB	P87C055BBP NB	0 to +70, 42-Pin Plastic Dual In-Line Package	3.5 to 12	1680



## Microcontroller for television and video (MTV) 83C055/87C055

## BLOCK DIAGRAM



## Microcontroller for television and video (MTV) 83C055/87C055

## PIN DESCRIPTIONS

MNEMONIC	PIN NO. DIP	TYPE	NAME AND FUNCTION
VCLK1	28	I	<b>Video Clock 1:</b> Input for the horizontal timing reference for the On Screen Display facility. VCLK1 and VCLK2 are intended to be used with an external LC circuit to provide an on-chip oscillator. The period of the video clock is determined such that the width of a pixel in the On Screen Display is equal to the inter-line separation of the raster.
VCLK2	29	O	<b>Video Clock 2:</b> Output from the on-chip video oscillator.
HSYNC	26	I	<b>Horizontal Sync:</b> A dedicated input for a TTL-level version of the horizontal sync pulse. The polarity of this pulse is programmable; its trailing edge is used by the On Screen Display facility as the reference for horizontal positioning.
VSYNC	27	I	<b>Vertical Sync:</b> A dedicated input for a TTL-level version of the vertical sync pulse. The polarity of this pulse is programmable, and either edge can serve as the reference for vertical timing.
VID2:0	22–24	O	<b>Digital Video bus:</b> Three totem pole outputs comprising digital RGB (or other color encoding) from the On Screen Display facility. The polarity of these outputs is controlled by a programmable register bit.
VCTRL	25	O	<b>Video Control:</b> A totem-pole output indicating whether the On Screen Display facility is currently presenting active video on the VID2:0 outputs. This signal should be used to control an external multiplexer (mixer) between normal video and the video derived from VID2:0. The polarity of this outputs is controlled by a programmable register bit.
BF	30	O	<b>Background/Foreground:</b> A totem-pole output which, when VCTRL is active, indicates whether the current video data represents a foreground (low) or background (high) dot in a character. This signal can be used to reduce the intensity of the background color and thus emphasize the text. If a 40-pin version of this part is ever produced, BF will not be pinned out.
P0.0-P0.7	1–8	I/O	<p><b>Port 0:</b> An 8-bit open-drain bidirectional port. Port 0 pins that have ones written to them float, and in that state can be used as high-impedance inputs. The port 0 pins can also serve as outputs from the high-precision Pulse Width Modulator (TDAC) and seven of the eight lower-precision Pulse Width Modulator functions. For each PWM block, a register bit controls whether the corresponding pin is controlled by the block or by port 0; port 0 controls the pin immediately after a Reset. Regardless of how each pin is controlled, it can be externally pulled up as high as +12V±5%, and the state of the pin can be read from the Port 0 register by the program.</p> <p><b>V<sub>PP</sub> (P0.0)</b> – This pin receives the 12V programming supply voltage during EPROM programming.</p> <p><b>PROG (P0.1)</b> – This pin receives the programming pulses during EPROM programming.</p> <p><b>ASEL (P0.2)</b> – Input which indicates which bits of the EPROM address are applied to port 2.</p> <p><b>TDAC (P0.0)</b> – This is the output for the 14-bit high-precision PWM.</p> <p><b>PWM1–7 (P0.1–P0.7)</b> – Outputs for the 6-bit PWMs 1 through 7.</p>
P1.0-P1.3	9–12	I/O	<p><b>Port 1:</b> A 4-bit open-drain bidirection port. Port 1 pins that have ones written to them float, and in that state can be used as high-impedance inputs. P1.3 can also serve as the eighth lower-precision Pulse Width Modulator output (PWM0), and can be externally pulled up as high as +12V±5%. P1.2:0 have optional alternate use as ADI2:0, inputs to the Software A/D conversion facility. If a 40-pin version of this part is ever produced, P1.3/PWM0 will not be pinned out.</p> <p>Any of the port 1 pins are driven low if the corresponding port register bit is written as 0, or, for P1.3 only, if the TDAC module presents a 0. The state of the pin can always be read from the port register by the program.</p> <p><b>ADI0–2 (P1.0–P1.2)</b> – Inputs for the software A/D facility.</p> <p><b>PWM0 (P1.3)</b> – Output for the PWM0 6-bit PWM.</p>
P2.0-P2.7	20–13	I/O	<p><b>Port 2:</b> An 8-bit open-drain bidirectional port. Port 2 pins that have ones written to them float, and in that state can be used as high-impedance inputs. P2.3:0 have high current capability (10 mA at 0.5V) for LEDs.</p> <p>Any of the port 2 pins are driven low if the port register bit is written as 0. The state of the pin can always be read from the port register by the program.</p>
P3.0-P3.7	34–42	I/O	<p><b>Port 3:</b> An 8-bit open-drain bidirectional port. Port 3 pins that have ones written to them float, and in that state can be used as high-impedance inputs. P3.0, P3.4, and P3.7 can be externally pulled up as high as +12V±5%, while P3.5 and P3.6 have 10mA drive capability. Some of the port 3 pins can also serve alternate functions, as follows:</p> <p><b>INT1 (P3.1)</b> – External Interrupt 1.</p> <p><b>T0 (P3.2)</b> – Timer 0 external input.</p> <p><b>INT0 (P3.3)</b> – External Interrupt 0.</p>



## Microcontroller for television and video (MTV) 83C055/87C055

PIN DESCRIPTIONS (Continued)

SYMBOL MNEMONIC	PIN NO. DIP	TYPE	NAME AND FUNCTION
RST	33	I	<b>Reset:</b> If this pin is high for two machine cycles (24 oscillator periods) while the oscillator is running, the MTV is reset. Also, this pin is used as a serial input to enter a test or EPROM programming mode, as on the 87C751.
XTAL1	31	I	<b>Crystal 1:</b> Input to the inverting oscillator amplifier and clock generator circuit that provides the timing reference for all MTV logic other than the OSD facility. XTAL1 and XTAL2 can be used with a quartz crystal or ceramic resonator to provide an on-chip oscillator. Alternatively, XTAL1 can be connected to an external clock, and XTAL2 left unconnected.
XTAL2	32	O	<b>Crystal 2:</b> Output from the inverting oscillator amplifier.
V <sub>CC</sub>		I	<b>Power Supply:</b> This is the power supply for normal and power-down modes.
V <sub>SS</sub>		I	<b>Ground:</b> 0V reference.

Table 1. 8XC053/54/55 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB								LSB
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
DPTR	Data pointer (2 bytes)										
DPH	Data pointer high	83H									00H
DPL	Data pointer low	82H									00H
			AF	AE	AD	AC	AB	AA	A9	A8	
IE*	Interrupt enable	A8H	EA	—	—	EVS	ET1	EX1	ET0	EX0	0x000000B
OSAD	On-screen address	9A									
OSAT*	On-screen attributes	98									
OSDT	On-screen data	99	C7	C6	C5	C4	C3	C2	C1	C0	
OSCON*	On-screen display control	C0	Iv	Pv	Lv	Ph	Pc	Po	DH	BF <sub>e</sub>	
OSMOD	On-screen display mode	C1	Wc	—	Mode 1	Mode 0	—	SHM2	SHM1	SHM0	
OSORG	On-screen display origin	C2	HS4	HS3	HS2	HS1	HS0	VS2	VS1	VS0	
			87	86	85	84	83	82	81	80	
P0*	Port 0	80H	PWM7	PWM6	PWM5	PWM4	PWM3	PWM2	PWM1	TDAC	FFH
			97	96	95	94	93	92	91	90	
P1*	Port 1	90H	—	—	—	—	PWM0	ADI2	ADI1	ADI0	FFH
P2*	Port 2	A0H	A7	A6	A5	A4	A3	A2	A1	A0	FFH
			B7	B6	B5	B4	B3	B2	B1	B0	
P3*	Port 3	B0H	—	—	—	—	INT0	T0	INT1	—	FFH
PCON	Power control	87H	—	—	—	—	GF1	GF0	PD	—	0xxxxxxxB



Microcontroller for television and video (MTV) 83C055/87C055

Table 1. 8XC053/54/55 Special Function Registers (Continued)

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program status word	D0H	CY	AC	F0	RS1	RS0	OV	—	P	00H
PWM0	Lo-res pulse width modulators	D4									
PWM1	Lo-res pulse width modulators	D5									
PWM2	Lo-res pulse width modulators	D6									
PWM3	Lo-res pulse width modulators	D7									
PWM4	Lo-res pulse width modulators	DC									
PWM5	Lo-res pulse width modulators	DD									
PWM6	Lo-res pulse width modulators	DE									
PWM7	Lo-res pulse width modulators	DF									
SAD*	D/A and voltage comparator	D8	VHI	CH1	CH0	St	SAD3	SAD2	SAD1	SAD0	
SP	Stack pointer	81H									07H
TDACH	Hi-res pulse width modulators	D3									
TDACL	Hi-res pulse width modulators	D2									
TCON*	Timer control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	
TH0	Timer high 0	8CH									00H
TH1	Timer high 1	8DH									00H
TL0	Timer low 0	8AH									00H
TL1	Timer low 1	8BH									00H
TMOD	Timer mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H

\* Bit addressable

ROM CODE SUBMISSION

When submitting a ROM code for the 83C055, the following must be specified:

1. The 16k byte user ROM program.
2. The OSD ROM space.

This information can be submitted in an 87C055, or in two EPROMs (2764), or electronically on the ROM Code Bulletin Board (see your local sales office for the number).

ROM CODE SUBMITTAL REQUIREMENTS

ADDRESS	CONTENT	COMMENT
000H to 3FFFF (83C055)	DATA	User ROM data
C000H to CFFFF	OSD	On-Screen Display character table

## Microcontroller for television and video (MTV) 83C055/87C055

## PROGRAMMING THE OSD EPROM

## Overview

The OSD EPROM space starts at location C000H and ends at location CFFFH. However, not all locations within this space are used, due to the addressing scheme of the OSD.

The start location of the next character can be calculated by adding 40H to the start location of the previous character. For example, character #1 starts at C000H; then characters 2, 3, and 4 start at C040H, C080H, and C0C0H, respectively.

## Character Description

Each character is 14 bits wide by 18 lines high.

A character is split about a vertical axis into two sections, UPPER and LOWER. Each section contains 7 bits of the character, such

that the LOWER section contains 1–7 and the UPPER section contains bits 8–14.

**NOTE:** During programming and verification, each section is programmed using bytes of DATA. The MSB of the DATA is not used; however, the MSB location physically exists, and so will program and verify.

The LOWER section of the character is programmed when the LSB of the program address equals 0, and the UPPER section when the LSB equals 1.

## Character Programming

An example of an OSD character bit map, and the program DATA to obtain that character is shown in Table 2.

## OSD EPROM Bit Map

The mapping for the full OSD EPROM is shown in Table 3.

## Example

To program the character given above into the first character location of the OSD EPROM would require the following address/DATA sequence:

C000/00H;	C001/00H;	C002/00H;
C003/00H;	C004/0CH;	C005/1EH;
C006/0CH;	C007/1EH;	C008/0CH;
C009/1EH;	C00A/0CH;	C00B/1EH;
C00C/0CH;	C00D/1EH;	C00E/0CH;
C00F/1EH;	C010/7CH;	C011/1FH;
C012/7CH;	C013/1FH;	C014/7CH;
C015/1FH;	C016/0CH;	C017/1EH;
C018/0CH;	C019/1EH;	C01A/0CH;
C01B/1EH;	C01C/0CH;	C01D/1EH;
C01E/0CH;	C01F/1EH;	C020/00H;
C021/00H;	C022/00H;	C023/00H;

Table 2. Example of an OSD Character Bit Map

CHARACTER BIT MAP		PROGRAM DATA	
	UPPER LOWER	UPPER LOWER	
	←→←→		
	11111		
	43210987654321		
Line 1 →	00000000000000	X0000000	X0000000
Line 2 →	00000000000000	X0000000	X0000000
Line 3 →	00111100001100	X0011110	X0001100
Line 4 →	00111100001100	X0011110	X0001100
Line 5 →	00111100001100	X0011110	X0001100
Line 6 →	00111100001100	X0011110	X0001100
Line 7 →	00111100001100	X0011110	X0001100
Line 8 →	00111100001100	X0011110	X0001100
Line 9 →	00111100001100	X0011110	X0001100
Line 10 →	00111100001100	X0011110	X0001100
Line 11 →	00111111111100	X0011111	X1111101
Line 12 →	00111111111100	X0011111	X1111101
Line 13 →	00111111111100	X0011111	X1111101
Line 14 →	00111111111100	X0011111	X1111101
Line 15 →	00111100001100	X0011110	X0001100
Line 16 →	00111100001100	X0011110	X0001100
Line 17 →	00111100001100	X0011110	X0001100
Line 18 →	00111100001100	X0011110	X0001100
	00111100001100	X0011110	X0001100
	00111100001100	X0011110	X0001100
	00000000000000	X0000000	X0000000
	00000000000000	X0000000	X0000000

## NOTE:

X can be 0 or 1, and will program and verify correctly.

## Microcontroller for television and video (MTV) 83C055/87C055

Table 3. OSD EPROM Bit Map

CHARACTER NO.	ADDRESS	CHARACTER LINE NO.	COMMENTS
1	C000	1	Lower byte
	C001	1	Upper byte
	C002	2	Lower byte
	C003	2	Upper byte
	C004	•	•
	C005	•	•
	C006	•	•
	C007	•	•
	C008	•	•
	C009	•	•
	C00A	•	•
	C00B	•	•
	C00C	•	•
	C00D	•	•
	C00E	•	•
	C00F	•	•
	C010	•	•
	C011	•	•
	C012	•	•
	C013	•	•
	C014	•	•
	C015	•	•
	C016	•	•
	C017	•	•
	C018	•	•
	C019	•	•
	C01A	•	•
	C01B	•	•
	C01C	•	•
	C01D	•	•
	C01E	•	•
	C01F	•	•
	C020	•	•
	C021	•	•
	C022	•	•
	C023	•	•
	C024-C03F	Unused	
2	C040-C063	1-18	
	C064-C07F	Unused	
3	C080-C0A3	1-18	
	C0A4-C0BF	Unused	
•	•	•	•
•	•	•	•
•	•	•	•
62	CFC0-CFE3	1-18	NEWLINE
	CFE4-CFFF	Unused	
63	CFC0-CFE3	1-18	BSPACE
	CFE4-CFFF	Unused	
64	CFC0-CFE3	1-18	SPLITBSPACE
	CFE4-CFFF	Unused	

**NOTE:**

Locations 62, 63, and 64 should be programmed to 0's.

**COMPARISON TO THE 80C51**

The elements of the MTV are shown in the Block Diagram. The features of the MTV are identical to those of the 80C51, except as noted herein.

**Pinout and Testing**

Since neither data nor program memory is externally expandable on the MTV, the 80C51 pins ALE, EA, and PSEN are not implemented on the MTV.

**I/O Ports**

On both the 80C51 and the MTV, port 0 is open-drain, but on the 80C51 it can be used for external memory expansion while on the MTV its alternate use is for Pulse Width Modulated outputs.

On the 80C51, port 1 is 8 bits, is mostly unallocated (general purpose), and is quasi-bidirectional (that is, having a weak pullup transistor that can be overdriven). On the MTV it is a 4-bit open-drain port, and includes alternate uses for analog inputs and a PWM output.

On the 80C51, port 2 is quasi-bidirectional and can be used for external memory expansion; on the MTV, port 2 is open-drain and unallocated.

On the 80C51, port 3 is quasi-bidirectional and all eight bits have alternate uses. On the

MTV, three port 3 bits have some of the same alternate uses as on the 80C51 but not necessarily on the same pins, while five pins are open-drain and unallocated.

**Idle Mode**

The idle mode is not implemented on the MTV.

**Power-Down Mode**

The power-down mode is not implemented on the MTV. The PCON register has the following format:

PCON	7	6	5	4	3	2	1	0
	—	—	—	—	GF1	GF0	—	—

**Interrupts**

The interrupt facilities of the MTV differ from those of the 80C51 as follows:

1. Since there is no serial port, there are no interrupts nor control bits relating to this interrupt. The interrupts and their vector addresses are as follows:

Event	Program Memory Address
Reset	000
External INTO	003
Timer 0	00B
External INT1	013
Timer 1	01B
VSync Start	023

2. The VSYNC input used by the On Screen Display facility can generate an interrupt. The active polarity of the pulse is programmable, as described in a later section. The interrupt occurs at the leading edge of the pulse.

3. External Interrupt 1 is modified so that an interrupt is generated when the input switches in either direction (on the 80C51, there is a programmable choice between interrupt on a negative edge or a low level on INT1). This facility allows for software pulse-width measurement handling of a remote control.

4. The IP register is not used, and the IE register is similar to that on the 80C51:

IE	7	6	5	4	3	2	1	0
	EA	—	—	EVS	ET1	EX1	ET0	EX0

## Microcontroller for television and video (MTV) 83C055/87C055

## Six-Bit PWM DACs

The structure of these modules is shown in Figure 2. First, the basic MCU clock is divided by 4 to get a waveform that clocks a 6-bit counter which is common to all the PWMs, including the 14-bit one. This divided clock is hereafter called the PWM counter clock.

Each PWM block has a special function register PWMn arranged as follows:

PWM0-PWM7							
7	6	5	4	3	2	1	0
PWE	—	PV5	PV4	PV3	PV2	PV1	PV0

If the PWE bit for a particular PWM block is 1, the block is active and controls its assigned port pin; if PWE is 0 the corresponding port pin is controlled by the port. The "value" field (PV5 ... PV0) of each PWM register is compared to (the LS 6 bits of) the common counter. When the value matches, the output FF is cleared, so that the output pin is driven low. When the value rolls over to zero, the output FF is set, so that the output pin is released. Thus the output waveform has a fixed period of 64 PWM counter clocks; its duty cycle is determined by PWMn.5:0.

Three of the nine total PWMs operate as described above; for three others, both the rising and falling edges of the output are delayed by one PWM clock; for the remaining three, both edges are delayed by two PWM clocks. This feature reduces the radio-frequency emission that would otherwise occur when the counter rolled over to zero and all nine open-drain outputs were released.

## 14-Bit PWM DAC (TDAC)

This feature was partially described in the preceding section. As shown in Figure 4, the 6-bit counter used for the lower precision PWMs is in fact the least significant part of a 14-bit counter used for this facility. The nature of the counter is such that it can achieve a stable output value through its MSB, and the value can propagate through logic like that shown in Figure 4, and the logic output can be stable within one period of the PWM counter clock (e.g., 250 ns) if edge-triggered logic is used to capture the logic output, or within one phase of the PWM counter clock (e.g., 125 ns) if a phase of the PWM counter clock is used to capture the logic output. For cost and die-size reasons, it is preferable that the TDAC counter be a ripple counter.

This feature is controlled by two special function registers:

TDACL							
7	6	5	4	3	2	1	0
TD7	TD0	TD1	TD2	TD3	TD4	TD5	TD6

TDACH							
7	6	5	4	3	2	1	0
TDE	—	TD13	TD12	TD11	TD10	TD9	TD8

When software wishes to change the 14-bit value (TD0 – TD13), it should first write TDACL and then write TDACH. Alternatively, if the required precision of the duty cycle is satisfied by 6 bits or less, software can simply write TDACH. Note from Figure 4 that this block includes an "extra" 14-bit latch between TDACL/H and the comparator and other logic. The programmed value is clocked into the operative latch when the 7 low-order bits of the counter roll over to zero, provided that the software is not in the midst of loading a new 14-bit value (that is, it is not between writing TDACL and writing TDACH).

In a similar fashion to the lower-precision PWMs, this facility has an output FF that is set when the lower 7 bits of the counter overflow/wrap. The *more* significant 7 bits of the operative latch's programmed value are compared for equality against the *less* significant 7 bits of the counter, and the output FF is cleared when they match. Thus this output has a fixed period of 128 PWM counter clocks, and the duty cycle is determined by the programmed value.

For the higher-precision aspect of this feature, the 7 more-significant bits of the counter are used in a logic block with the 7 less-significant bits of the programmed value. The 7th LSB (binary value 64) of the programmed value is ANDed with the 7th MSB (128) of the counter, the 6th LSB of the value is ANDed with the counter's 6th and 7th MSBs being 10, and so on through the LSB of the programmed value being ANDed with the counter's 7MSBs being 100000. Then these 7 ANDed terms are ORed. If the result is true/1 at the time the 7 LSBs of the counter match the MSBs of the programmed value, the output is forced high for 1 (additional) PWM counter clock.

The result is that, if the value-64 bit of the 14-bit value is programmed to 1, every other cycle of 128 PWM counter clocks has its duty cycle stretched by one counter clock; if the value-32 bit is programmed to 1, every 4th cycle is stretched, and so on through, if the value-1 bit is programmed to 1, one cycle out of each 128 is stretched.

Assuming the external integrator can handle all this, the net effect is a PWM DAC that has the period of a 7-bit design (which makes the integrator easier and more feasible to design) with the accuracy of a 14-bit one. There is some question whether all of the least significant bits can be effectively integrated, or whether they simply act as a source of

ripple in the integrated voltage. An obvious prerequisite for such precision is that the load on the voltage must be very light, like a single op amp or comparator.

The TDAC feature differs from the corresponding features of predecessor parts in several ways:

1. The 14-bit value is functionally composed of major and minor portions of 7 bits each.
2. The 14-bit value is programmed as a contiguous multi-register value that can be manipulated straight-forwardly via arithmetic instructions.
3. As discussed for the 6-bit DACs, both of the preceding parts had a feature whereby the PWM output could be inverted, redundantly with complementing the 14-bit value. This feature has been eliminated.

## Software A/D Facility

This facility is shown in Figure 5. It represents an alternate use whereby any of the P1.0 through P1.2 pins can be selected as one input of a linear voltage comparator. The block includes one special function register:

SAD							
7	6	5	4	3	2	1	0
VHI	CH1	CH0	St	SAD3	SAD2	SAD1	SAD0

As shown in Figure 5 the other input of the comparator is connected to a 4-bit D/A that is controlled by the 4 LSBs of the SAD register, producing a reference voltage nominally 0.15625V to 4.84375V by steps of 0.3125V. The output of the comparator (high/low) can be read by the program as the MSB of the register, which is bit addressable.

The St bit should be written as 1 in order to initiate a voltage comparison. After writing St=1, the program should include intervening instructions totalling at least six machine cycles (72 CLK periods or 6 microseconds at 12MHz), before the instruction that accesses and tests VHI.

The chan field controls which pin, if any, is connected to this facility:

CH1	CH0	pin
0	0	none
0	1	P1.0
1	0	P1.1
1	1	P1.2

Port 1 has open-drain drivers which will not materially affect an analog voltage as long as any and all pins used for software A/D measurement have corresponding ones in the port register.

## Microcontroller for television and video (MTV) 83C055/87C055

	ADDRESS TYPE			USE
	DIRECT	BIT	REGISTER	
<b>DATA MEMORY</b>	00-07		R0-R7	On-chip RAM (R0-7 if PSW.4-3 = 00)
	08-0F		R0-R7	On-chip RAM (R0-7 if PSW.4-3 = 01)
	10-17		R0-R7	On-chip RAM (R0-7 if PSW.4-3 = 10)
	18-1F		R0-R7	On-chip RAM (R0-7 if PSW.4-3 = 11)
	20	07-00		On-chip RAM
	21-2E	77-08		On-chip RAM
	2F	7F-78		On-chip RAM
	30-7F			On-chip RAM
	80	87-80	P0	Port 0
	81		SP	Stack Pointer
<b>SPECIAL FUNCTION REGISTERS</b>	82		DPL	Data Pointer LSBYTE
	83		DPH	Data Pointer MSBYTE
	87		PCON	Power Control
	88	8F-88	TCN	Timer Control
	89		TMOD	Timer Mode
	8A		TL0	Timer 0 LSBYTE
	8B		TL1	Timer 1 LSBYTE
	8C		TH0	Timer 0 MSBYTE
	8D		TH1	Timer 1 MSBYTE
	90	97-90	P1	Port 1
	98	9F-98	OSAT	On Screen Attributes
	99		OSDT	On Screen Data
	9A		OSAD	On Screen Address
	A0	A7-A0	P2	Port 2
	A8	AF-A8	IE	Interrupt Enable
	B0	B7-B0	P3	Port 3
	C0	C7-C0	OSCON	On Screen Display Control
	C1		OSMOD	On Screen Display Mode
	C2		OSORG	On Screen Display Origin
	C3		RAMCHR	For Test Use Only
	C4		RAMATT	For Test Use Only
	D0	D7-D0	PSW	Program Status Word
	D2		TDACL	Hi-Res Pulse Width Modulator
	D3		TDACH	Hi-Res Pulse Width Modulator
	D4-D7		PWM0-3	Lo-Res Pulse Width Modulators
	D8	DF-D8	SAD	D/A and Voltage Comparator
	DC-DF		PWM4-7	Lo-Res Pulse Width Modulators
	E0	E7-E0	A	Accumulator
	F0	F7-F0	B	Register

Figure 1. Data Memory and Special Function Registers on the MTV



## Microcontroller for television and video (MTV) 83C055/87C055

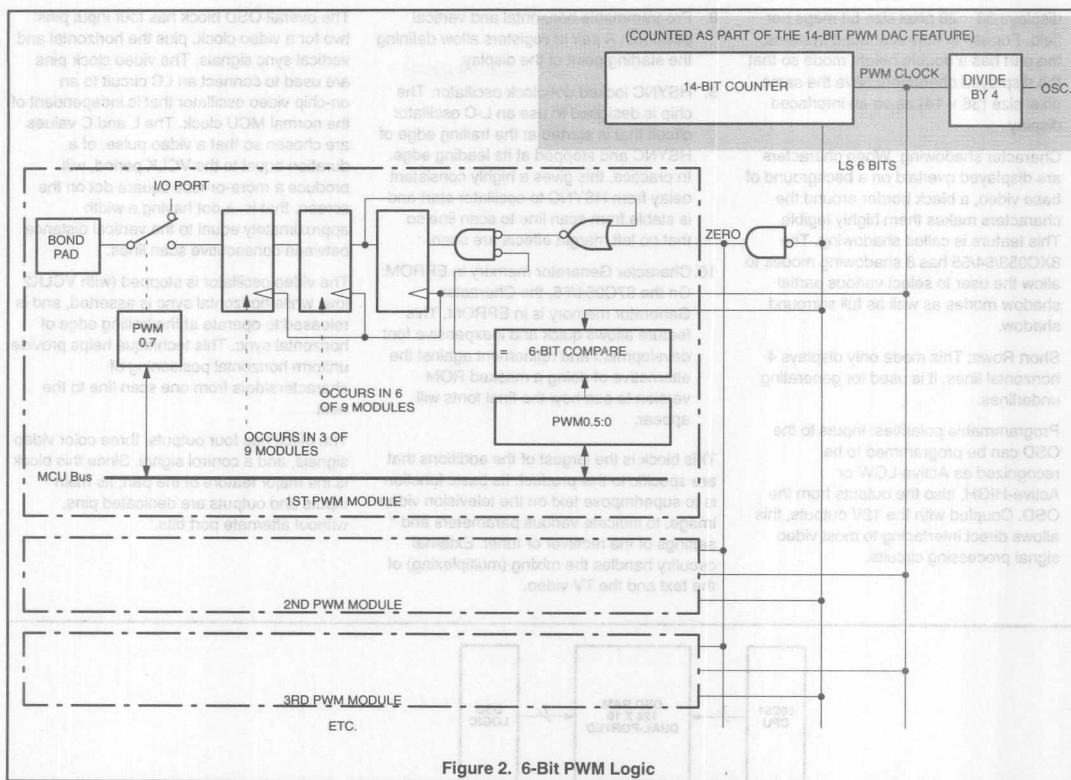


Figure 2. 6-Bit PWM Logic

### On Screen Display (OSD) Module

Figure 3 is a conceptual drawing of the OSD block on the 8XC053/54/55.

It shows the CPU writing into the  $128 \times 10$  display RAM, which is dual-ported to allow the CPU to write into it at any time, including when it is being read out by the OSD logic.

The 10-bit wide data coming out of the display RAM is used to access the appropriate character in the Character Generator memory (6-bits) and to specify character and display control functions (4-bits). Timing for the OSD is controlled by the HSYNC, VSYNC, and dot clock inputs.

The 8XC053/54/55 features an advanced OSD function with some unique features:

1. **User-definable display format:** The OSD does not restrict the user to a fixed number of lines with a fixed number of characters per line as other competing alternatives do.

Using a fixed number of lines restricts the generation of displays that can be differentiated from others that use the

same chip and places limits on screen content.

Using a fixed number of characters per line wastes display RAM if a line has less than the full number of displayable characters (it has to be padded with non-visible characters).

The OSD on the 8XC053/54/55 defines a control character (New Line) that has the same function as a Carriage Return and Line Feed. When the OSD circuitry fetches this character from display RAM it stops displaying further characters, waits for the next horizontal scan line, and starts displaying the next character in display RAM after the New Line character was received. The number of lines is thus up to the user, within the limits of the display and memory, as are the number of characters per line. This allows far better control of the appearance of the On Screen Display.

2. **Dual-Ported Display RAM:** The OSD has a true display RAM instead of a character line buffer. This display RAM is dual ported to allow updating the display RAM at any time instead of having to wait for a vertical retrace. Vertical Sync interrupts are supported if flicker-free updates are required.
3. **Colors selectable by character:** Characters can be displayed on a background of the base video or a programmable background color. The background color is selectable by word and the choice of background (base video/user programmed color) by character.
4. **Programmable character size:** Normal characters are displayed as  $18 \times 14$  bit-pixels. In an interlaced display, 2 fields are displayed so that one actually sees a  $36 \times 14$  pixel size character. The part has a double height and width mode which



## Microcontroller for television and video (MTV) 83C055/87C055

displays  $36 \times 28$  pixel size bit maps per field. For use in non-interlaced systems, the part has a double height mode so that the displayed characters have the same pixel size ( $36 \times 14$ ) as on an interlaced display.

5. Character shadowing: When characters are displayed overlaid on a background of base video, a black border around the characters makes them highly legible. This feature is called shadowing. The 8XC053/54/55 has 8 shadowing modes to allow the user to select various partial shadow modes as well as full surround shadow.
6. Short Rows: This mode only displays 4 horizontal lines. It is used for generating underlines.
7. Programmable polarities: Inputs to the OSD can be programmed to be recognized as Active-LOW or Active-HIGH, also the outputs from the OSD. Coupled with the 12V outputs, this allows direct interfacing to most video signal processing circuits.

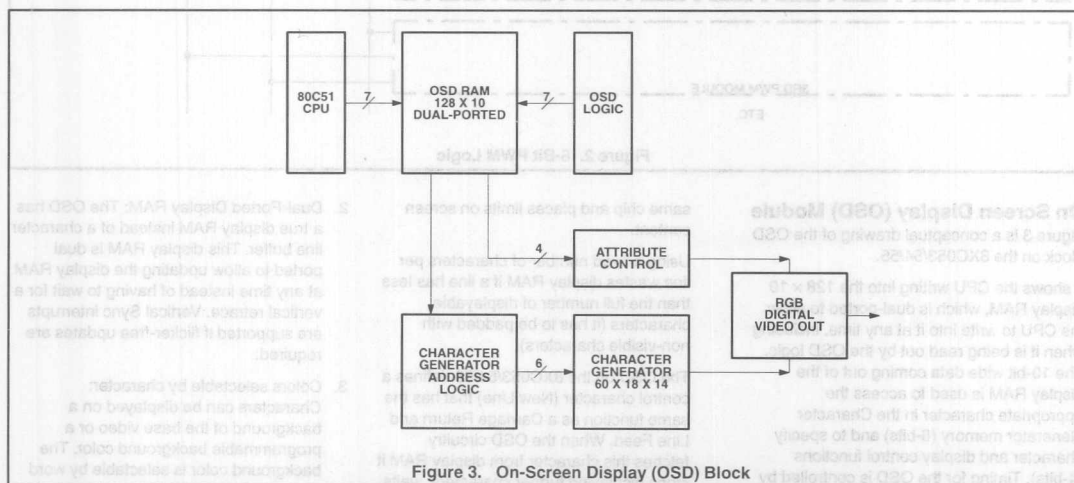
8. Programmable horizontal and vertical positions: A pair of registers allow defining the starting point of the display.
9. HSYNC locked dot-clock oscillator: The chip is designed to use an L-C oscillator circuit that is started at the trailing edge of HSYNC and stopped at its leading edge. In practice, this gives a highly consistent delay from HSYNC to oscillator start and is stable from scan line to scan line so that no left margin effects are seen.
10. Character Generator memory in EPROM: On the 87C054/55, the Character Generator memory is in EPROM. This feature allows quick and inexpensive font development and refinement against the alternative of doing a masked ROM version to see how the final fonts will appear.

This block is the largest of the additions that are specific to this product. Its basic function is to superimpose text on the television video image, to indicate various parameters and settings of the receiver or tuner. External circuitry handles the mixing (multiplexing) of the text and the TV video.

The overall OSD block has four input pins: two for a video clock, plus the horizontal and vertical sync signals. The video clock pins are used to connect an LC circuit to an on-chip video oscillator that is independent of the normal MCU clock. The L and C values are chosen so that a video pulse, of a duration equal to the VCLK period, will produce a more-or-less square dot on the screen, that is, a dot having a width approximately equal to the vertical distance between consecutive scan lines.

The video oscillator is stopped (with VCLK2 low) while horizontal sync is asserted, and is released to operate at the trailing edge of horizontal sync. This technique helps provide uniform horizontal positioning of characters/dots from one scan line to the next.

The block has four outputs, three color video signals, and a control signal. Since this block is the major feature of the part, its main inputs and outputs are dedicated pins, without alternate port bits.



**Figure 3. On-Screen Display (OSD) Block**

## Microcontroller for television and video (MTV) 83C055/87C055

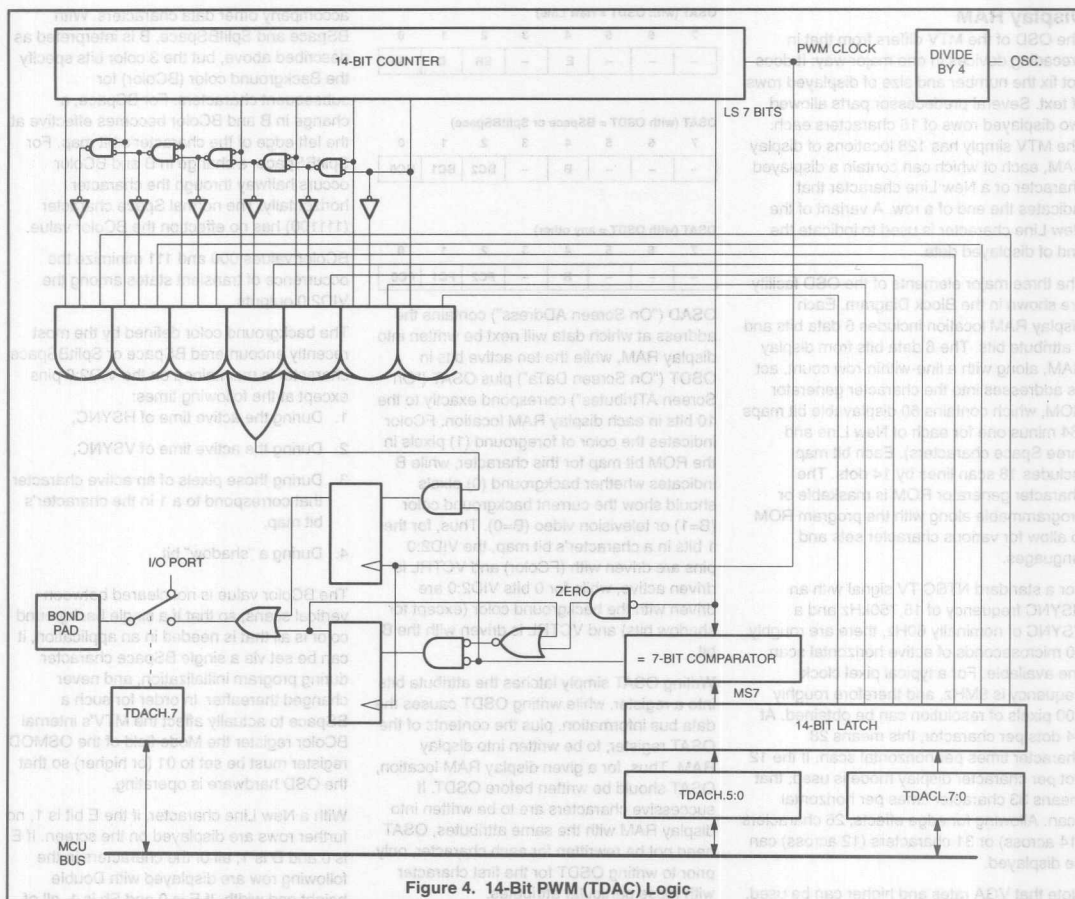


Figure 4. 14-Bit PWM (TDAC) Logic

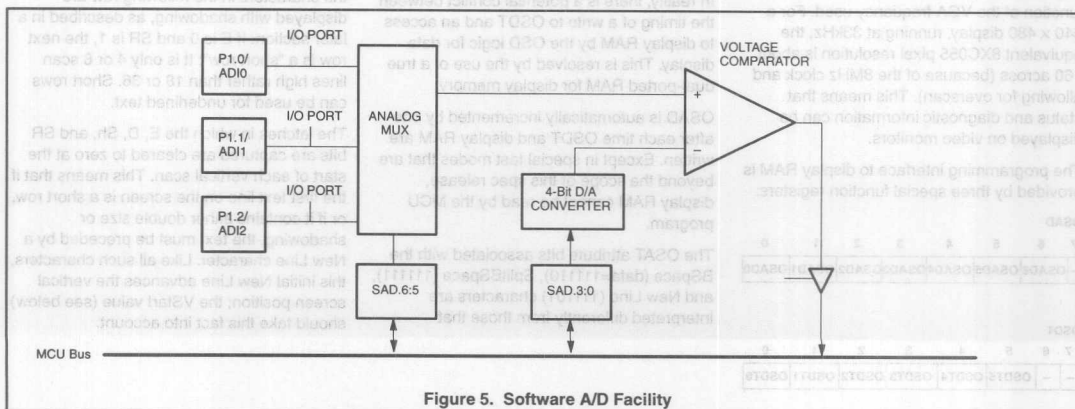


Figure 5. Software A/D Facility

## Microcontroller for television and video (MTV) 83C055/87C055

### Display RAM

The OSD of the MTV differs from that in preceding devices in one major way: It does not fix the number and size of displayed rows of text. Several predecessor parts allowed two displayed rows of 16 characters each. The MTV simply has 128 locations of display RAM, each of which can contain a displayed character or a New Line character that indicates the end of a row. A variant of the New Line character is used to indicate the end of displayed data.

The three major elements of the OSD facility are shown in the Block Diagram. Each display RAM location includes 6 data bits and 4 attribute bits. The 6 data bits from display RAM, along with a line-within-row count, act as addresses into the character generator ROM, which contains 60 displayable bit maps (64 minus one for each of New Line and three Space characters). Each bit map includes 18 scan lines by 14 dots. The character generator ROM is maskable or programmable along with the program ROM to allow for various character sets and languages.

For a standard NTSC TV signal with an HSYNC frequency of 15,750kHz and a VSYNC of nominally 60Hz, there are roughly 50 microseconds of active horizontal scan line available. For a typical pixel clock frequency is 8MHz, and therefore roughly 400 pixels of resolution can be obtained. At 14 dots per character, this means 28 character times per horizontal scan. If the 12 dot per character display mode is used, that means 33 character times per horizontal scan. Allowing for edge effects, 26 characters (14 across) or 31 characters (12 across) can be displayed.

Note that VGA rates and higher can be used. The minimum character dot size will be a function of the VGA frequency used. For a 640 x 480 display, running at 33kHz, the equivalent 8XC055 pixel resolution is about 160 across (because of the 8MHz clock and allowing for overscan). This means that status and diagnostic information can be displayed on video monitors.

The programming interface to display RAM is provided by three special function registers:

OSAD							
7	6	5	4	3	2	1	0
—	OSAD6	OSAD5	OSAD4	OSAD3	OSAD2	OSAD1	OSAD0

OSDT							
7	6	5	4	3	2	1	0
—	—	OSDT5	OSDT4	OSDT3	OSDT2	OSDT1	OSDT0

OSAT (with OSDT = New Line)

7	6	5	4	3	2	1	0
—	—	—	E	—	SR	D	Sh

OSAT (with OSDT = BSpace or SplitBspace)

7	6	5	4	3	2	1	0
—	—	—	B	—	BC2	BC1	BC0

OSAT (with OSDT = any other)

7	6	5	4	3	2	1	0
—	—	—	B	—	FC2	FC1	FC0

OSAD ("On Screen Address") contains the address at which data will next be written into display RAM, while the ten active bits in OSDT ("On Screen Data") plus OSAT ("On Screen Attributes") correspond exactly to the 10 bits in each display RAM location. FColor indicates the color of foreground (1) pixels in the ROM bit map for this character, while B indicates whether background (0) pixels should show the current background color (B=1) or television video (B=0). Thus, for the 1 bits in a character's bit map, the VID2:0 pins are driven with (FColor) and VCTRL is driven active, while for 0 bits VID2:0 are driven with the background color (except for shadow bits) and VCTRL is driven with the B bit.

Writing OSAT simply latches the attribute bits into a register, while writing OSDT causes the data bus information, plus the contents of the OSAT register, to be written into display RAM. Thus, for a given display RAM location, OSAT should be written before OSDT. If successive characters are to be written into display RAM with the same attributes, OSAT need not be rewritten for each character, only prior to writing OSDT for the first character with those particular attributes.

In reality, there is a potential conflict between the timing of a write to OSDT and an access to display RAM by the OSD logic for data display. This is resolved by the use of a true dual-ported RAM for display memory.

OSAD is automatically incremented by one after each time OSDT and display RAM are written. Except in special test modes that are beyond the scope of this spec release, display RAM cannot be read by the MCU program.

The OSAT attribute bits associated with the BSpace (data=111110), SplitBspace (111111), and New Line (111101) characters are interpreted differently from those that

accompany other data characters. With BSpace and SplitBspace, B is interpreted as described above, but the 3 color bits specify the Background color (BColor) for subsequent characters. For BSpace, a change in B and BColor becomes effective at the left edge of the character's bit map. For SplitBspace, a change in B and BColor occurs halfway through the character horizontally. The normal Space character (111100) has no effect on the BColor value.

BColor values 000 and 111 minimize the occurrence of transient states among the VID2:0 outputs.

The background color defined by the most recently encountered BSpace or SplitBspace character is maintained on the VID2:0 pins except at the following times:

1. During the active time of HSYNC,
2. During the active time of VSYNC,
3. During those pixels of an active character that correspond to a 1 in the character's bit map,
4. During a "shadow" bit.

The BColor value is not cleared between vertical scans, so that if a single background color is all that is needed in an application, it can be set via a single BSpace character during program initialization, and never changed thereafter. In order for such a BSpace to actually affect the MTV's internal BColor register the Mode field of the OSMOD register must be set to 01 (or higher) so that the OSD hardware is operating.

With a New Line character, if the E bit is 1, no further rows are displayed on the screen. If E is 0 and D is 1, all of the characters in the following row are displayed with Double height and width. If E is 0 and Sh is 1, all of the characters in the following row are displayed with shadowing, as described in a later section. If E is 0 and SR is 1, the next row is a "short row": It is only 4 or 8 scan lines high rather than 18 or 36. Short rows can be used for underlined text.

The latches in which the E, D, Sh, and SR bits are captured are cleared to zero at the start of each vertical scan. This means that if the first text line on the screen is a short row, or if it contains either double size or shadowing, the text must be preceded by a New Line character. Like all such characters, this initial New Line advances the vertical screen position; the VStart value (see below) should take this fact into account.

## Microcontroller for television and video (MTV) 83C055/87C055

### Other OSD Registers

A number of changes in the OSD architecture have reduced the number of other special function registers involved in the feature, below the number needed with predecessor devices:

1. The elimination of certain options such as 4, 6, or 8X character sizes and alternate use of two of the video outputs.
2. The moving of certain other options from central registers to display RAM, such as foreground color codes and background selection.

#### OSCON

7	6	5	4	3	2	1	0
IV	Pv	Lv	Ph	Pc	Po	DH	BFe

The IV bit is the interrupt flag for the OSD feature. It is set by the leading edge of the VSYNC pulse, and is cleared by the hardware when the VSYNC interrupt routine is vectored to. It can also be set or cleared by software writing a 1 or 0 to this bit.

#### NOTE

It is theoretically possible that a VSYNC interrupt could be missed, or an extra one generated, if OSCON is read, then modified internally (e.g., in Ac), and the result written back to OSCON. However, none of the other bits in OSCON are reasonable candidates for dynamic change. Special provisions are included in the MTV logic so that IV will not be changed by a single "read-modify-write" instruction such as SETB or CLR, unless the instruction specifically changes IV.

A 0 (1) in Pv designates that the VSYNC input is high-active (low-active). One effect of this bit is that the VID2:0 and VCTRL outputs are blocked (held at black/inactive) during the active time of VSYNC. The IV bit is set on the leading edge of the VSYNC pulse; thus Pv controls whether the OSD interrupt occurs in response to a high-to-low or low-to-high transition on VSYNC.

A 0 (1) in Lv designates that the leading edge (active level) of VSYNC, as defined by Pv, clears the state counter that is used to determine the vertical start of on-screen data. In effect, Lv=0(1) says that the leading (trailing) edge of VSYNC is the time reference for the video field.

A 0 (1) in Ph designates that the HSYNC input is high-active (low-active).

A 0 (1) in Pc designates that a high (low) on the VCTRL output means "show the color on VID2:0".

A 0 (1) in Po designates that a 0 (1) internal to the MTV corresponds to a low on one of the VID2:0 pins. This control bit is needed only because the Shadowing feature needs to generate black pixels without reference to a register value: Internally, the 3-bit code 000 always designates black.

If DH is 1, character sizes are doubled vertically but not horizontally. This feature allows the MTV to be used in "improved definition" systems that are not interlaced. The vertical doubling imposed by DH does not affect the VStart logic described below: It operates in HSync units regardless of DH or D.

If BFe is 1, the BF output tracks whether each bit in displayed characters is a foreground bit (low) or a background bit (high). If BFe is 0, the BF pin remains high.

#### OSORG

7	6	5	4	3	2	1	0
HS4	HS3	HS2	HS1	HS0	VS2	VS1	VS0

The HStart field (HS4 – HS0) defines the left end (start) of all of the on-screen character rows, as a multiple of four VCLKs. Active display begins 4(HStart)+1 VCLKs plus one single-sized character width after the trailing edge of HSYNC. Counting variations in Wc, there may be 17 to 143 VCLKs from the end of HSYNC to the start of the first character of each row.

The VStart field (VS2 – VS0) defines the top (start) of the first on-screen character row, as a multiple of four HSYNC pulses. Active display begins 4(VStart)-1 HSYNCS after the field's time reference point, a range of 3 to 31. Subsequent character rows occur directly below the first, such that the last scan line of one row is directly followed by the first scan line of the next row. Successive New Line characters (with or without the Short Row designation) can be used to vertically separate text rows on the screen.

Neither the HStart nor VStart parameter is affected by the D line attribute that is used to display double-sized characters.

#### OSMOD

7	6	5	4	3	2	1	0
Wc	-	Mode1	Mode0	-	SHM2	SHM1	SHM0

If the mode bits (Mode 1, Mode 0) are 00, the OSD feature is disabled. The VCLK oscillator is disabled, VID2:0 are set to black, and

VCTRL is held inactive. This is the mode to which the MTV OSD logic is reset. A direct transition from this mode to active display (1x) would result in undefined operation and visual effects for the duration of the current video field (until the next VSYNC).

If the mode is 01, the VCLK oscillator is enabled and the OSD logic operates normally internally, but VID2:0 are set to black and VCTRL is held inactive. The OSD feature can be toggled between this state and 1x as desired to achieve real-time special effects such as "vertical wiping."

Mode 10 represents normal OSD operation. Active characters can be shown against TV video (for characters with B=0) or (for characters with B=1) against a background of the color defined as an attribute of BSpace and SplitBspace characters.

In mode 11, characters can be displayed but all of the receiver's normal video is inhibited by holding VCTRL asserted throughout the active portion of each scan line. Since VID2:0 are driven with the current background color during this time, except during the foreground portion of displayed characters, this produces text against a solid background. This mode is useful for extensive displays that require user concentration.

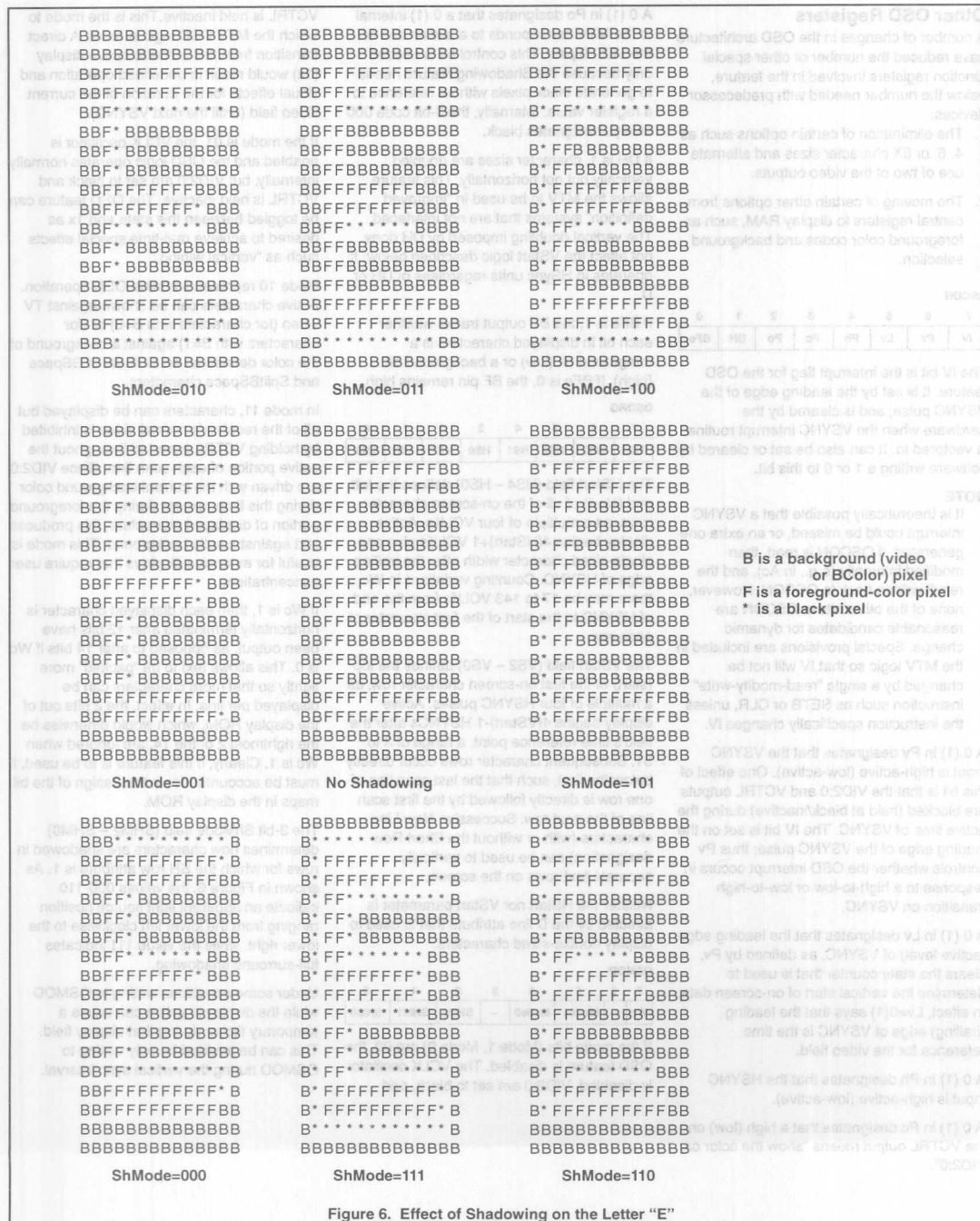
If Wc is 1, then each displayed character is horizontally terminated after 12 bits have been output, as opposed to after 14 bits if Wc is 0. This allows text to be "packed" more tightly so that more characters can be displayed per line. In effect, the 2 bits out of the display ROM, which would otherwise be the rightmost 2 of the 14, are ignored when Wc is 1. Clearly, if this feature is to be used, it must be accounted for in the design of the bit maps in the display ROM.

The 3-bit ShMode field (SHM2 – SHM0) determines how characters are shadowed in rows for which the SH row attribute is 1. As shown in Figure 6, the values 000-110 indicate an apparent light source position ranging from the lower left clockwise to the lower right, while the value 111 indicates full-surround shadowing.

Under some conditions writing to OSMOD while the display is active can cause a temporary flicker during that display field. This can be avoided by only writing to OSMOD during the vertical sync interval.



## Microcontroller for television and video (MTV) 83C055/87C055



## DC ELECTRICAL CHARACTERISTICS

 $V_{CC} = 5V \pm 10\%$ ,  $T_{amb} = 0^\circ C$  to  $+70^\circ C$ 

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT	NOTES
			MIN	MAX		
$V_{IL}$	Input low voltage		-0.5	$0.2V_{CC}-0.1$	V	
$V_{IL1}$	Input low voltage (VSYNC, HSYNC)		-0.5	$0.15 \times V_{CC}$	V	
$V_{IH1}$	Input high voltage (P1.2:0, P2.7:0, P3.6:5, P3.3:1)		$0.2V_{CC}+0.9$	$V_{CC}+0.5$	V	
$V_{IH2}$	Input high voltage (port 0, P1.3, P3.7, P3.4, P3.0)		$0.2V_{CC}+0.9$	12.6	V	
$V_{IH3}$	Input high voltage (VSYNC, HSYNC)		$0.67 \times V_{CC}$	$V_{CC} + 0.5V$	V	
$V_{IH} - V_{CC}$	Input high voltage (port 0, P1.3, P3.7, P3.4, P3.0) with respect to $V_{CC}$			8	V	1
$V_{IH}$	Input high voltage (XTAL1, VCLK1, RST)		$0.7V_{CC}$	$V_{CC}+0.5$	V	
$V_{OL1}$	Output low voltage (P2.7:0, P3.6:5)	$I_{OL} = 10mA$		0.5	V	5
$V_{OL2}$	Output low voltage (TDAC, PWM0:7)	$I_{OL} = 700\mu A$		0.5	V	2
$V_{OL3}$	Output low voltage (all other outputs)	$I_{OL} = 1.6mA$		0.45	V	
$V_{OH}$	Output high voltage (port 1, VID2:0, VCTRL, BF)	$I_{OH} = -60\mu A$	2.4		V	
$R_{RST}$	Reset pulldown resistor		50	300	k $\Omega$	
$C_{IO}$	Pin capacitance (except P0.0 and P0.7)	Test freq = 1MHz, $T_{amb} = 25^\circ C$		10	pF	4
$I_{CC}$	Normal mode supply current	$V_{CC} = 5.5V$		30	mA	3
HYS	Hysteresis (VSYNC, HSYNC)		0.8		V	

## NOTES:

- This maximum applies at all times, including during power switching, and must be accounted for in power supply design. During a power-on process, the +12 volt source used for external pullup resistors should not precede the  $V_{CC}$  of the MTV up their respective voltage ramps by more than this margin, nor, during a power-down process, should  $V_{CC}$  precede +12V down their respective voltage ramps by more than this margin.
- The specified current rating applies when any of these pins is used as a Pulse Width modulated output. For use as a port output, the rating is as given subsequently.
- $I_{CC}$  measured with OSD block initialized and Reset remaining low.
- The capacitance of pins P0.0 and P0.7 for the 87C055 exceeds 10pF. P0.0 is 40pF maximum, while P0.7 is 20pF maximum.
- No more than 6 (any 6) of these 10 high current outputs may be used at the  $V_{OL1}$  ( $I_{OL} = 10mA$ ) specification. The other 4 should comply with the  $V_{OL3}$  specification ( $I_{OL} = 1.6mA$ ).

The Vpp signal may now be driven to the Vpp level, placing the 87C055 in the verify mode. Port 5 is now used as an output port. After four machine cycles (48 clock periods), the contents of the address in the EPROM array will appear on Port 5.

The next programming cycle may now be initiated by placing the address information at the inputs of the multiplexed buffers, driving the Vpp pin to the Vpp voltage level, providing the byte to be programmed to Port 5 and resuming the 56 programming bytes on the PGM pin, driving Vpp back down to the Vpp level and verifying the byte. (See Table 4.)

**Programming Operation**

Figures 8 and 9 show the timing diagram for the programmable cycle. RESET should initially be held high for at least two machine cycles. Port 1 (PGM) and P0.0 (Vpp) will be at Vpp as a result of the RESET operation. At this point, these pins function as normal dual-in-line package VO pins and the programming equipment may now drive these pins low. However, prior to sending the 10-bit code on the RESET pin, the programming equipment should drive these pins high (Vpp). The RESET pin may now be used as the serial data input for the data stream which places the 87C055 in the programming mode. Data bits are sampled during the clock high time and this should only change during the time that the clock is low. Following transmission of the last data bit, the RESET pin should be held low.

Port 5 is used as the address input for the byte to be programmed and occupies port 5's high and low components of the address bus. Multiplexing of these address components is performed using the ABSEL input. The user should drive the ABSEL input high and drive port 5 with the high order bits of the address. ABSEL should remain high for at least 10 clock cycles. ABSEL may then be driven low which latches the high order bits of the address internally. The high order address remains on port 5 for at least two clock cycles after ABSEL is driven low. Port 5 may then be driven with the low byte of the address. The low address will be internally latched 10 clock cycles later. The address will remain stable provided that the low byte placed on port 5 is held stable and ABSEL is kept low. Note: ABSEL needs to be pushed high only to change the high byte of the address.



## Microcontroller for television and video (MTV) 83C055/87C055

### AC ELECTRICAL CHARACTERISTICS

$V_{CC} = 5V \pm 10\%$ ,  $T_{amb} = 0^\circ C$  to  $+70^\circ C$

SYMBOL	PARAMETER	TENTATIVE LIMITS		UNIT	NOTES
		MIN	MAX		
$1/t_{CLCL}$	XTAL Frequency	6	12	MHz	1
$t_{CHCX}$	XTAL1 Clock high time	20		ns	2
$t_{CLCX}$	XTAL1 Clock low time	20		ns	2
$t_{CLCH}$	XTAL1 Clock rise time		20	ns	2
$t_{CLCL}$	XTAL1 Clock fall time	5	20	ns	2
$1/t_{VCLCL}$	VCLK Frequency	5	8	MHz	
$ t_{VCOH} - t_{VCOL} $	Rise vs. fall time skew on any one of VID2:0, VCTRL, BF		40	ns	3
$ t_{VCOH1} - t_{VCOH2} $	Rise time skew between any two of VID2:0, VCTRL, BF		30	ns	3
$ t_{VCOL1} - t_{VCOL2} $	Fall time skew between any two of VID2:0, VCTRL, BF		30	ns	3

#### NOTES:

1. The MTV is tested at its maximum XTAL frequency, but not at any other (lower) rate.
2. These parameters apply only when an external clock signal is used.
3. These parameters assume equal loading at  $C_L = 100pF$ , for all the referenced outputs. These parameters are specified but not tested.

### PROGRAMMING CONSIDERATIONS

#### EPROM Characteristics

The 87C055 is programmed by using a modified Quick-Pulse Programming algorithm similar to that used for devices such as the 87C51. It differs from these devices in that a serial data stream is used to place the 87C751 in the programming mode.

Figure 7 shows a block diagram of the programming configuration for the 87C055. Port pin P0.0 is used as the programming voltage supply input ( $V_{PP}$  signal). Port pin P0.1 is used as the program (PGM) signal. This pin is used for the 25 programming pulses.

Port 2 is used as the address input for the byte to be programmed and accepts both the high and low components of the eleven bit address. Multiplexing of these address components is performed using the ASEL input. The user should drive the ASEL input high and then drive port 2 with the high order bits of the address. ASEL should remain high for at least 13 clock cycles. ASEL may then be driven low which latches the high order bits of the address internally. The high address should remain on port 2 for at least two clock cycles after ASEL is driven low. Port 2 may then be driven with the low byte of the address. The low address will be internally stable 13 clock cycles later. The address will remain stable provided that the low byte placed on port 2 is held stable and ASEL is kept low. **Note:** ASEL needs to be pulsed high only to change the high byte of the address.

Port 3 is used as a bidirectional data bus during programming and verify operations. During programming mode, it accepts the byte to be programmed. During verify mode, it provides the contents of the EPROM location specified by the address which has been supplied to Port 2.

The XTAL1 pin is the oscillator input and receives the master system clock. This clock should be between 1.2 and 6MHz.

The RESET pin is used to accept the serial data stream that places the 87C055 into various programming modes. This pattern consists of a 10-bit code with the LSB sent first. Each bit is synchronized to the clock input, X1.

#### Programming Operation

Figures 8 and 9 show the timing diagrams for the program/verify cycle. RESET should initially be held high for at least two machine cycles. P0.1 (PGM) and P0.0 ( $V_{PP}$ ) will be at  $V_{OH}$  as a result of the RESET operation. At this point, these pins function as normal quasi-bidirectional I/O ports and the programming equipment may pull these lines low. However, prior to sending the 10-bit code on the RESET pin, the programming equipment should drive these pins high ( $V_{IH}$ ). The RESET pin may now be used as the serial data input for the data stream which places the 87C055 in the programming mode. Data bits are sampled during the clock high time and thus should only change during the time that the clock is low. Following transmission of the last data bit, the RESET pin should be held low.

Next the address information for the location to be programmed is placed on port 2 and ASEL is used to perform the address multiplexing, as previously described. At this time, port 1 functions as an output.

A high voltage  $V_{PP}$  level is then applied to the  $V_{PP}$  input (P0.0). (This sets Port 1 as an input port). The data to be programmed into the EPROM array is then placed on Port 3. This is followed by a series of programming pulses applied to the PGM/ pin (P0.1). These pulses are created by driving P0.1 low and then high. This pulse is repeated until a total of 25 programming pulses have occurred. At the conclusion of the last pulse, the PGM/ signal should remain high.

The  $V_{PP}$  signal may now be driven to the  $V_{OH}$  level, placing the 87C055 in the verify mode. (Port 3 is now used as an output port). After four machine cycles (48 clock periods), the contents of the addressed location in the EPROM array will appear on Port 3.

The next programming cycle may now be initiated by placing the address information at the inputs of the multiplexed buffers, driving the  $V_{PP}$  pin to the  $V_{PP}$  voltage level, providing the byte to be programmed to Port 3 and issuing the 26 programming pulses on the PGM/ pin, bringing  $V_{PP}$  back down to the  $V_{OH}$  level and verifying the byte. (See Table 4.)

## Microcontroller for television and video (MTV) 83C055/87C055

### Erase Characteristics

Erase of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent erasure. **For this and secondary effects, it is recommended that an opaque label be**

**placed over the window.** For elevated temperature or environments where solvents are being used, apply Kapton tape Flourless part number 2345-5 or equivalent.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 angstroms) to an integrated dose of at least 15W-s/cm<sup>2</sup>. Exposing the EPROM to an ultraviolet lamp of 12,000μW/cm<sup>2</sup> rating for 20 to 39 minutes, at a distance of about 1 inch, should be sufficient.

Erase leaves the array in an all 1s state.

### Reading Signature Bytes

The signature bytes are read by the same procedure as a normal verify of locations 30H and 31H, except that the serial code indicated in Table 3 for reading signature bytes should be used. The values are:

30H = 15H indicates manufactured by Philips  
31H = 4BH indicates 87C055

**Table 4. Implementing Program/Verify Modes**

OPERATION	SERIAL CODE	P0.1 (PGM/)	P0.0 (V <sub>PP</sub> )
Program user EPROM	286H	—*	V <sub>PP</sub>
Verify user EPROM	286H	V <sub>IH</sub>	V <sub>IH</sub>
Read Signature Bytes	280H	V <sub>IH</sub>	V <sub>IH</sub>

#### NOTE:

\* Pulsed from V<sub>IH</sub> to V<sub>IL</sub> and returned to V<sub>IH</sub>.

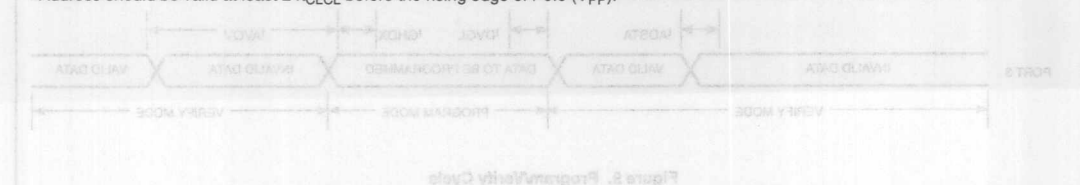
### EPROM PROGRAMMING AND VERIFICATION

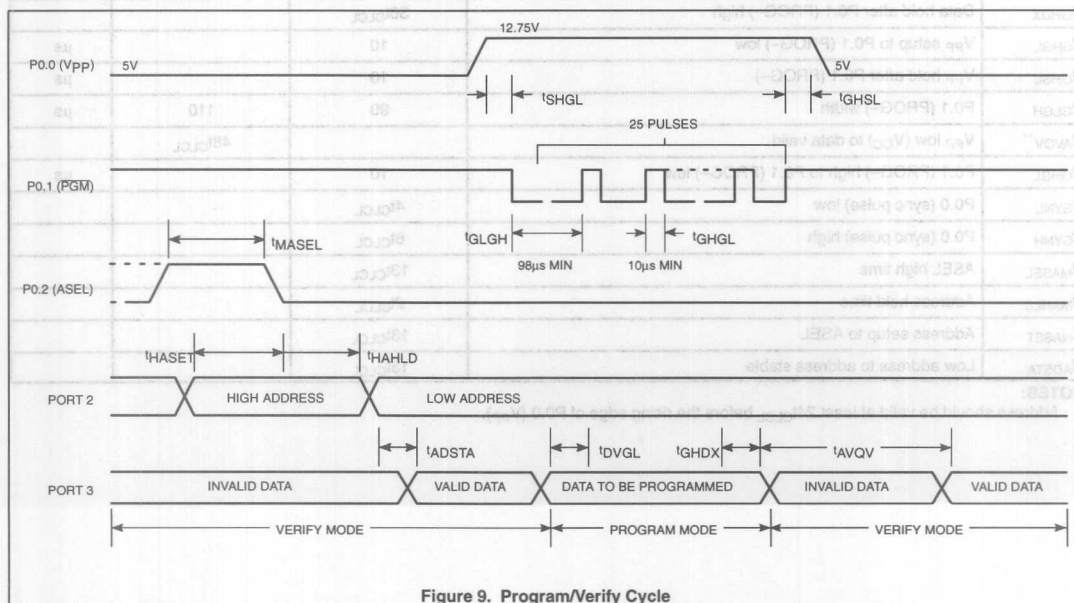
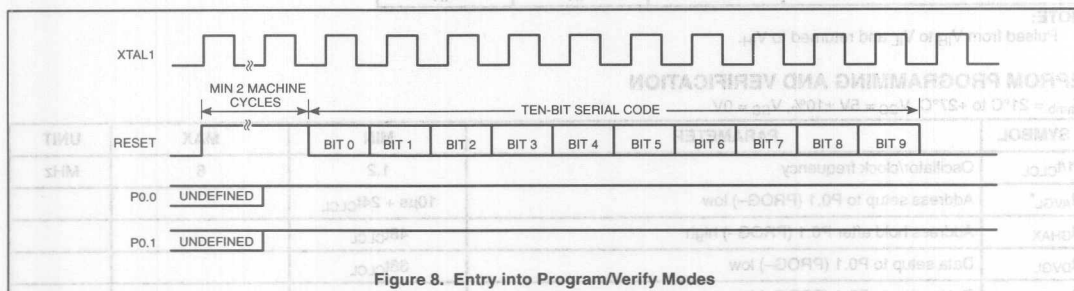
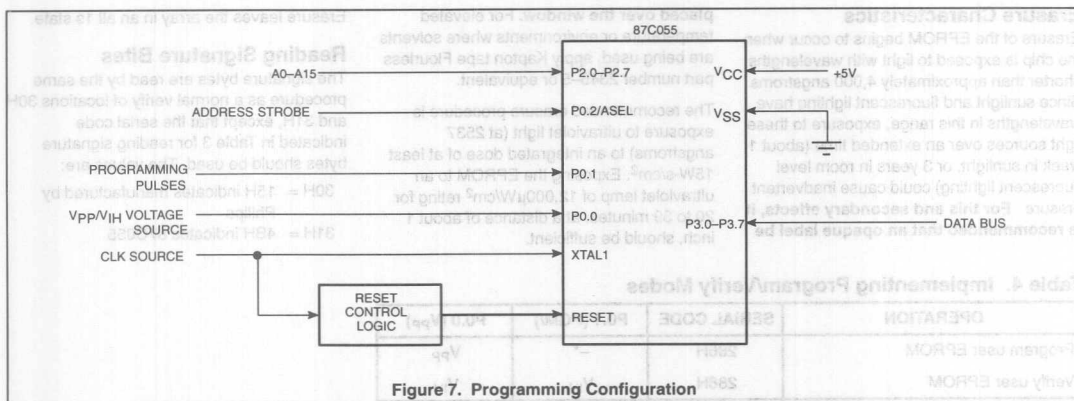
T<sub>amb</sub> = 21°C to +27°C, V<sub>CC</sub> = 5V ±10%, V<sub>SS</sub> = 0V

SYMBOL	PARAMETER	MIN	MAX	UNIT
1/t <sub>CLCL</sub>	Oscillator/clock frequency	1.2	6	MHz
t <sub>AVGL</sub> *	Address setup to P0.1 (PROG-) low	10μs + 24t <sub>CLCL</sub>		
t <sub>GHAX</sub>	Address hold after P0.1 (PROG-) high	48t <sub>CLCL</sub>		
t <sub>DVGL</sub>	Data setup to P0.1 (PROG-) low	38t <sub>CLCL</sub>		
t <sub>DVGL</sub>	Data setup to P0.1 (PROG-) low	38t <sub>CLCL</sub>		
t <sub>GHDX</sub>	Data hold after P0.1 (PROG-) high	36t <sub>CLCL</sub>		
t <sub>SHGL</sub>	V <sub>PP</sub> setup to P0.1 (PROG-) low	10		μs
t <sub>GHSL</sub>	V <sub>PP</sub> hold after P0.1 (PROG-)	10		μs
t <sub>GLGH</sub>	P0.1 (PROG-) width	90	110	μs
t <sub>AVQV</sub> **	V <sub>PP</sub> low (V <sub>CC</sub> ) to data valid		48t <sub>CLCL</sub>	
t <sub>GHGL</sub>	P0.1 (PROG-) high to P0.1 (PROG-) low	10		μs
t <sub>SYNL</sub>	P0.0 (sync pulse) low	4t <sub>CLCL</sub>		
t <sub>SYNH</sub>	P0.0 (sync pulse) high	8t <sub>CLCL</sub>		
t <sub>MASEL</sub>	ASEL high time	13t <sub>CLCL</sub>		
t <sub>MAHLD</sub>	Address hold time	2t <sub>CLCL</sub>		
t <sub>HASET</sub>	Address setup to ASEL	13t <sub>CLCL</sub>		
t <sub>ADSTA</sub>	Low address to address stable	13t <sub>CLCL</sub>		

#### NOTES:

\* Address should be valid at least 24t<sub>CLCL</sub> before the rising edge of P0.0 (V<sub>PP</sub>).





# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

## P83CL168/167/268/267

### 1. INTRODUCTION

The P83CL168/167/268/267 single chip microcontroller is an 80C51 CPU based MicroController designed for medium high to high end TV control applications. It features a Philips 80CL51 CPU which is object code compatible with the industry standard 80C51. It is manufactured in an advanced CMOS technology featuring low power consumption.

The differences between these four devices are shown in Table 1.

For the rest of the specification we will use P83CL168 as the generic term to describe the function of this family.

The P83CL168 also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit handling capabilities. Multiply & divide instructions are implemented by hardware with a cycle time of 4 $\mu$ s (XTAL=12MHz).

The instruction set consists of over 100 instructions; 49 one-byte, 46 two-byte and 16 three-byte.

The P83CL168 contains 16K bytes of ROM, 256 bytes of RAM, 21 general I/O port lines, 28 function combined I/O lines.

The **derivative functions** dedicated for TV applications are:

- AFC (A/D converter), remote control pre-processing circuit,
- PWM's for analog control, and voltage synthesized tuning (VST),
- on screen display function (OSD).

There are also two serial I/O's available, the Philips standard I<sup>2</sup>C (two sets) and the high speed 8-bit 3-wire serial I/O and an 8-bit counter (T2) for AC line frequency or HSYNC pulse counting.

### 1.1 P80CL51 CPU Core Features

- Full static 80C51 CPU
- 8 bit CPU, ROM, RAM, I/O's in a single package
- ROM max memory space : 64K (but P83CL168 has no external ROM access capability)
- RAM max memory space : 64K (but P83CL168 has no external RAM access capability)
- Two 16-bit timer/event counters
- On-chip oscillator suitable for quartz crystal or ceramic resonator
- Thirteen source, thirteen vector interrupt structure with two priority levels
- Enhanced architecture with:
  - non page orientated instructions
  - direct addressing
  - four eight byte RAM register banks
  - stack depth up to 128 bytes
  - multiply, divide, subtract and compare instructions

Table 1. Differences between P83CL168/167/268/267

	P83CL168	P83CL167	P83CL268	P83CL267
ROM	16K	16K	12K	12K
HIO <sup>1</sup>	YES	NO	YES	NO
RMC <sup>2</sup>	YES	NO	YES	NO
Pin 26 <sup>3</sup>	P57/RMC	P57	P57/RMC	P57
Pin 27 <sup>4</sup>	P40/D <sub>OUT</sub>	P40	P40/D <sub>OUT</sub>	P40
Pin 28 <sup>4</sup>	P41/CLK	P41	P41/CLK	P41
Pin 29 <sup>4</sup>	P42/D <sub>IN</sub>	P42	P42/D <sub>IN</sub>	P42

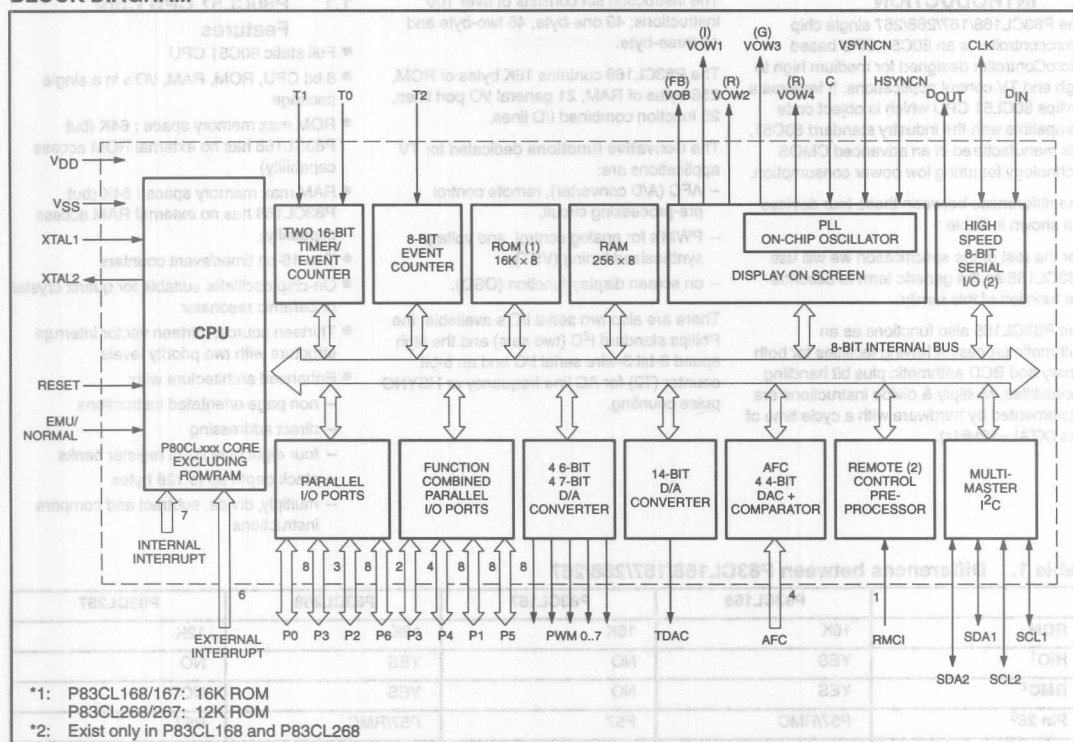
#### NOTES:

1. HIO, High speed I/O, 8-bit serial bus with maximum 1MHz baud rate. (see chapter 15)
2. RMC, ReMote Control pre-processor, a derivative function dedicated to preprocess I/R remote control signals. (see chapter 16)
3. Pin 26 is the combined function pin of RMC input and/or port line 57. Selection of either one function is controlled by a SFR P5SEL. (see chapter 9)
4. Pin 27, 28 and 29 are for HIO as well as for port lines. Selection is done by SFR P4SEL. (see chapter 9)

# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

## P83CL168/167/268/267

### BLOCK DIAGRAM





# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

## P83CL168/167/268/267

### 1.2 P83CL168 VST and OSD Derivative Features

- ROM/RAM
  - 16K × 8 system ROM, 256 × 8 system RAM on chip (no external memory access capability)
- PWMs
  - One 14-bit PWM output for VST (voltage synthesized tuning)
  - Four 6-bit PWM and four 7-bit PWM outputs (DACs for analog controls)
- AFC (A to D converter)
  - Four AFC inputs with 4 bit DAC and comparator (i.e., 4 channel of 4-bit A to D converter)
- LED driver port
  - Eight port lines (P0) with 10mA LED drive (@ <0.4V) capability
- Remote Control pre-processor
  - 8-bit counter with pulse width measurement capability & digital filter for noise discrimination.
  - Resolution of minimal pulse width is programmed by software (Fxtal/192, /384, /768, /1536)
  - Maximum noise discrimination pulse width programmed by software (Fxtal/96, /192, /384, /768)
  - Interrupt driven or polling by software.
- Serial I/O: 2 types
  3. Multimaster I<sup>2</sup>C (with two sets of SDA/SCL pins)
  4. High speed (1MHz baud rate maximum) 8-bit 3-wire serial I/O
- Timer/counter T2 – AC supply line frequency counter (e.g., real time clock application in TV) or HSYNC counter (e.g., no broadcasting signal detection)
  - 8-bit event counter driven by external input. Rising edge Schmitt trigger input

#### • On screen display function

- Programmable VSYNC/HSYNC active level
- Display RAM: 160 × 12 bit
- Display character fonts: 128 (126 customer font + 2 reserved)
- One vertical display starting position counter. 64 different starting positions controlled by software.
- One horizontal display starting position control for OSD. 64 different positions controlled by software.
- Character size: 4 different character size, line-by-line basis, 1 horizontal dot = 1 or 2 OSD clock period, 1 vertical dot = 1 or 2 horizontal scan lines. Individually controllable.
- Character matrix: 12 × 18 with no spacing between characters
- Foreground colors: 16, character-by-character basis
- Background/shadowing modes: Two primary modes: "TV mode" or "Monitor mode", frame basis. In each primary mode, three submodes are available.
  - Sub-mode 1: superimpose (no background),
  - sub-mode 2: north-west shadowing,
  - sub-mode 3: box shadowing, all these three are line basis.

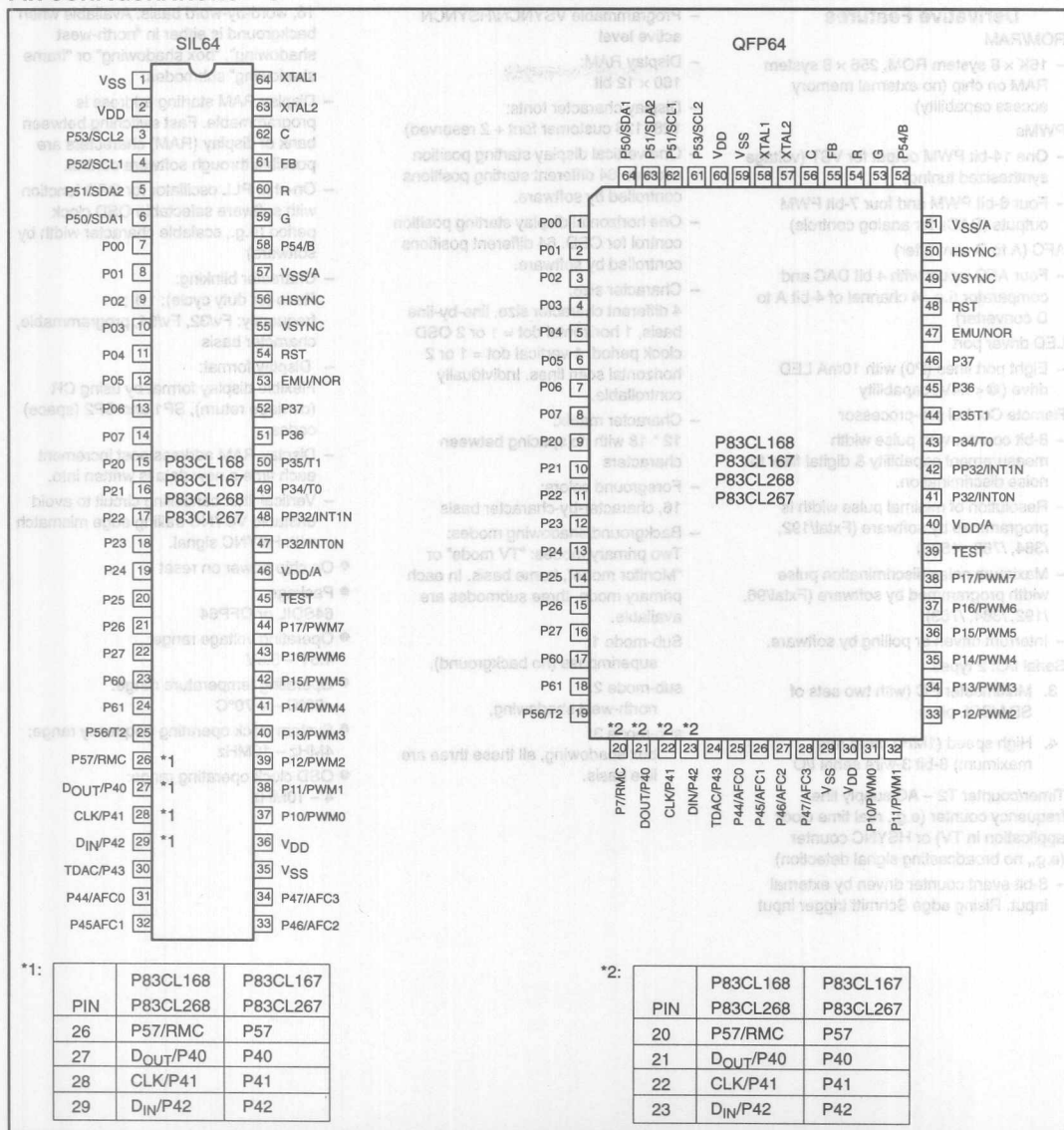
- Background colors: 16, word-by-word basis. Available when background is either in "north-west shadowing", "box shadowing" or "frame shadowing" submodes.
- Display RAM starting address is programmable. Fast switching between bank of display (RAM) characters are possible through software control.
- On-chip PLL oscillator for OSD function with software selectable OSD clock period (e.g., scalable character width by software)
- Character blinking: Ratio (or duty cycle): 1:1; frequency: Fv/32, Fv/64, programmable, character basis
- Display format: Flexible display format by using CR (carriage return), SP1 and SP2 (space) codes.
- Display RAM address post increment each time a new data is written into.
- Vertical jitter cancelling circuit to avoid unstable VSYNC trailing edge mismatch with HSYNC signal.
- On chip power on reset
- Package: 64SDIL or QFP64
- Operating voltage range: 4.5V – 5.5V
- Operating temperature range: –20°C – +70°C
- System clock operating frequency range: 4MHz – 12MHz
- OSD clock operating range: 4 – 10MHz

P83CL167	P83CL168	P83CL169
P83CL167	P83CL168	P83CL169
P83CL167	P83CL168	P83CL169
P83CL167	P83CL168	P83CL169
P83CL167	P83CL168	P83CL169
P83CL167	P83CL168	P83CL169
P83CL167	P83CL168	P83CL169
P83CL167	P83CL168	P83CL169

P83CL167	P83CL168	P83CL169
P83CL167	P83CL168	P83CL169
P83CL167	P83CL168	P83CL169
P83CL167	P83CL168	P83CL169
P83CL167	P83CL168	P83CL169
P83CL167	P83CL168	P83CL169
P83CL167	P83CL168	P83CL169
P83CL167	P83CL168	P83CL169



## PIN CONFIGURATIONS



# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

## 1.3 Pin Assignment

PIN NUMBER		PIN NAME	I/O	FUNCTION
SDIL	QFP			
2, 36	30, 60	V <sub>DD</sub>		Digital Power line, 4.5V – 5.5V
1, 35	29, 59	V <sub>SS</sub>		Digital Ground line
46	40	V <sub>DD</sub> /A		Analog Power line, 4.5V – 5.5V
57	51	V <sub>SS</sub> /A		Analog Ground line
3	61	P53/SCL2	I/O	I <sup>2</sup> C clock 2 (open drain) or port line 53 (pseudo bidirectional port)
4	62	P52/SCL1	I/O	I <sup>2</sup> C clock 1 (open drain) or port line 52 (pseudo bidirectional port)
5	63	P51/SDA2	I/O	I <sup>2</sup> C data 2 (open drain) or port line 51 (pseudo bidirectional port)
6	64	P50/SDA1	I/O	I <sup>2</sup> C data 1 (open drain) or port line 50 (pseudo bidirectional port)
14 – 7	8 – 1	P07 – P00	I/O	General I/O port (pseudo bidirectional port), also used in the emulation mode for control signals transfer in between bondout chip and the derivative
22 – 15	16 – 9	P27 – P20	I/O	General I/O port 2 (pseudo bidirectional port), also used in the emulation mode for control signals transfer in between bondout chip and the derivative
25	19	P56/T2	I/O	I/O port line 56 (pseudo bidirectional port) or 8-bit event counter input (Schmitt trigger)
26	30	P57/RMC	I/O	I/O port line 57 (pseudo bidirectional port) or Remote Control pre-processor input
29	23	D <sub>IN</sub> /P42	I/O	High speed serial I/O data input or port line 42 (pseudo bidirectional port)
28	22	CLK/P41	I/O	High speed serial I/O clock (bidirectional) or port line 41 (pseudo bidirectional port)
27	21	D <sub>OUT</sub> /P40	I/O	High speed serial I/O data output or port line 40 (pseudo bidirectional port)
30	24	TDAC/P43	I/O	14-bit PWM output (push-pull) or port line 43 (pseudo bidirectional port)
34 – 31	28 – 25	P47/AFC3 – P44/AFC0	I/O	I/O port line 47 – 44 (pseudo bidirectional port) or 4-bit AFC inputs (standard option)
44 – 41, 40 – 37	38 – 33	P17/PWM7 – P10/PWM0	I/O	I/O port line 17 – 10 (pseudo bidirectional port) or 7-bit PWM (PWM7 – 4) and 6-bit PWM (PWM3 – 0) output, also these pins P10, P11, P12 and P17 are used for external interrupt request if enabled by software
45	39	TEST	I	Test input (low) or normal operation (high)
52, 51	46, 45	P37, P36	I/O	I/O port line 37, 36, 31
50	44	P35/T1	I/O	I/O port line 35 or 16-bit timer/counter 1 I/O
49	43	P34/T0	I/O	I/O port line 34 or 16-bit timer/counter 0 I/O
48	42	P33/INT1N	I/O	I/O port line 33 or external interrupt 1 input
47	41	P32/INT0N	I/O	I/O port line 32 or external interrupt 0 input
53	47	EMU/NOR	I	Emulation mode (low) or normal mode (high) selection. Set to high in normal operation
54	48	RST	I	RESET input (active high)
56	50	HSYNC	I	TV horizontal sync input (for OSD synchronization)
55	49	VSYNC	I	TV vertical sync input (for OSD synchronization)
62	56	C	I/O	Capacitor input/output for OSD on chip PLL oscillator VCO
60	54	R	I/O	OSD Red color output
59	53	G	I/O	OSD Green color output
58	52	P54/B	I/O	I/O port line 54 or OSD Blue color output
61	56	FB	O	OSD fast blanking output
64, 63	58, 57	XTAL1/2	I/O	XTAL1, crystal input; XTAL2, crystal output
23, 24	17, 18	P60, P61	O	Output port line 60 and 61 (standard option only)

# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

## 2. MEMORY ORGANIZATION

The P83CL168 has 16K program memory (ROM) plus 256 bytes of data memory (RAM) on chip. The device has separate address spaces for program and data memory.

This device has NO external memory access capability such that the RD (read), WR (write), EA (external access), PSEN (read strobe) and ALE (address latch enable) signals are not bonded out.

## 2.1 Program Memory and Interrupt Vector

After reset the CPU begins execution at location 0000H. The interrupt vectors are located at the following address in ROM.

See also chapter 6 for interrupt level and function control details.

Priority (within level)	Source	Vector (address in HEX decimal)
PX0 (Highest)	External Interrupt 0 (INT0N)	0003H
S1	I <sup>2</sup> C	002BH
PX5	High speed 8-bit serial I/O	0053H
T0	Timer 0	000BH
PX6	Remote Control preprocessor	005BH
PX1	External Interrupt 1 (INT1H)	0013H
PX2	P10 (port line 10)	003BH
PX7	VS <sub>SYNC</sub> interrupt	0063H
T1	Timer 1	0B10H
PX3	P11 (port line 11)	0043H
PX8	T2 (AC frequency counter)	006BH
PX4	P12 (port line 12)	004BH
PX9 (Lowest)	P17 (port line 17)	0073H

Pin	Pin Name	Function
17, 18	P80, P81	Output port line 80 and 81 (standard option only)
28, 27	XTAL1, XTAL2	Crystal input; XTAL2, crystal output
26	P8	OSD test blanking output
25	P8A/B	VO port line 24 or OSD blue color output
23	G	OSD Green color output
24	R	OSD Red color output
26	C	Capacitor input/output for OSD on chip PLL oscillator VCO
29	VS <sub>SYNC</sub>	TV vertical sync input (for OSD synchronization)
30	HS <sub>SYNC</sub>	TV horizontal sync input (for OSD synchronization)
48	RESET	RESET input (active high)
47	EMULATOR	Emulation mode (low) or normal mode (high) selector. Set to high in normal operation
41	P8ANTON	VO port line 32 or external interrupt 6 input
42	P8ANT1H	VO port line 33 or external interrupt 7 input
43	P8AT0	VO port line 34 or 18-bit timer/counter 0 input
44	P8AT1	VO port line 35 or 18-bit timer/counter 1 input
45, 46	P8T, P82	VO port line 37, 38, 39
48	TEST	Test input (low) or normal operation (high)
49 - 52	P17/P16M0 - P17/P16M3	14-bit PWM (P16M0 - P16M3) output. Also these pins P10, P11, P12 and P17 are used for external interrupt request if enabled by software
53 - 54	P17/P16M0 - P17/P16M3	14-bit PWM (P16M0 - P16M3) output. Also these pins P10, P11, P12 and P17 are used for external interrupt request if enabled by software

# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

## 3. I/O FACILITY

### 3.1 I/O Ports

The P83CL168 has 49 I/O lines treated as 49 individual addressable bits or as 5 parallel 8-bit addressable ports, e.g., 0, 1, 2, 4, 5. Port 3 has 6 lines, Port 5 has 7 lines while port 6 has only two lines available.

Port 1, 3, 4 and 5 also performs the following alternate functions:

- Port 1: provides the output pins of the 6-bit/7-bit PWM's, as well as the external interrupt inputs (P10, P11, P12 & P17).
- Port 34, 35: provides 16-bit timer/counter 0 and 1's I/O.
- Port 32, 33: provide also the external interrupt input INT0N and INT1N.
- Port 40 – 42: provides the high speed 8-bit serial I/O's data input, output and clock pins.
- Port 43: provides the 14-bit PWM output (TDAC) for tuning voltage control.
- Port 44 – 47: provides the 4-bit AFC inputs.

- Port 50 – 53: provides the I<sup>2</sup>C data and clock I/O's.
- Port 54 – 55: provides the output of the OSD color B & I.
- Port 56: provides the I/O pin of the 8-bit event counter T2.
- Port 57: provides the remote control preprocessor input.

To enable the port 1, 3, 4 and/or 5 as alternate function, chapter 9 has the details. Each port consists of a latch (SFR P0 – P6), an output driver and an input buffer.

### 3.2 Port Options

Most of the I/O pins can be individually configured as one of the following port options (Figure 1, Figure 2, Figure 3). Port option selection table is shown in Table 2. **Option 1** — standard port; quasi-bidirectional I/O with pull-up. The strong booster pull-up transistor P1 is turned on for two oscillator periods after a 0-to-1 transition in the port latch. (Figure 1). To configure the port line

into OUTPUT port, write either 0 or 1 to its port latch (SFR bit). To configure this port line into an INPUT port, write an "1" to the port latch first before read the port pin.

**Option 2** — open drain; quasi-bidirectional I/O with n-channel pull-down. Use as an output requires the connection of an external pull-up resistor. (Figure 2). Use as an input requires to write an "1" to the port latch before read the port line.

**Option 3** — Push-pull; gives drive capability of the output in both polarities. This option permits the port to be used as an output only (Figure 3).

Individual mask selection of the post-reset state is available on any of the above pins. Make your selection by appending "S" or "R" to option 1, 2 or 3 above.

Option S : SET; after reset this pin is initialized to "HIGH".

Option R : RESET; after reset this pin is initialized to "LOW".

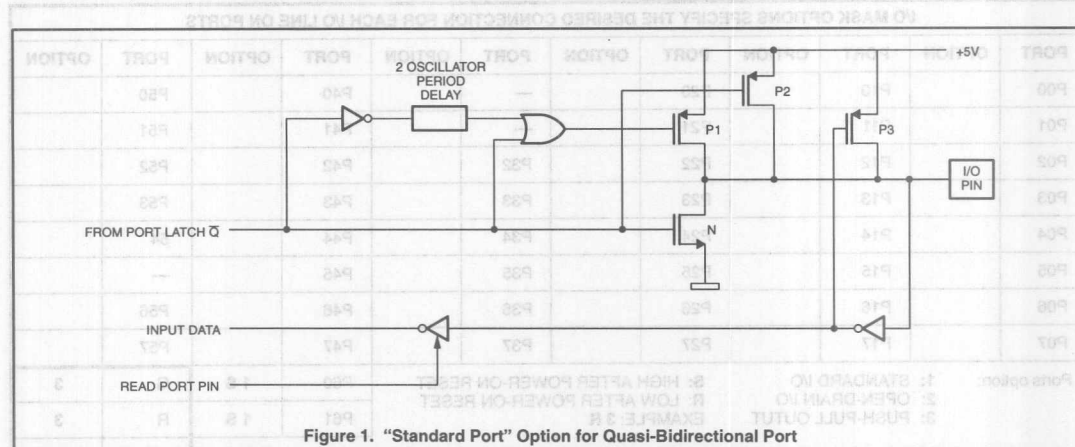


Figure 1. "Standard Port" Option for Quasi-Bidirectional Port

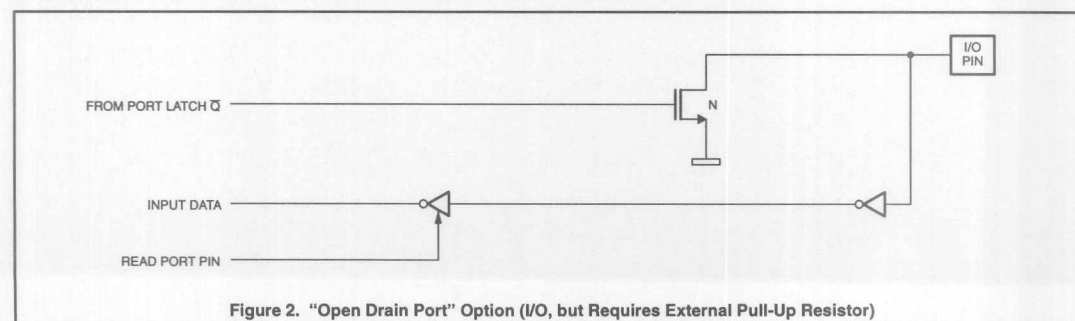


Figure 2. "Open Drain Port" Option (I/O, but Requires External Pull-Up Resistor)

# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

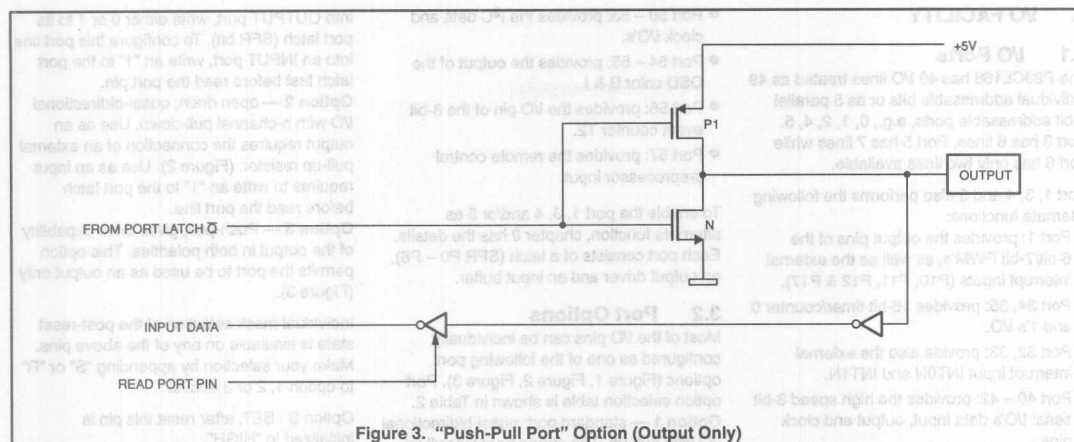
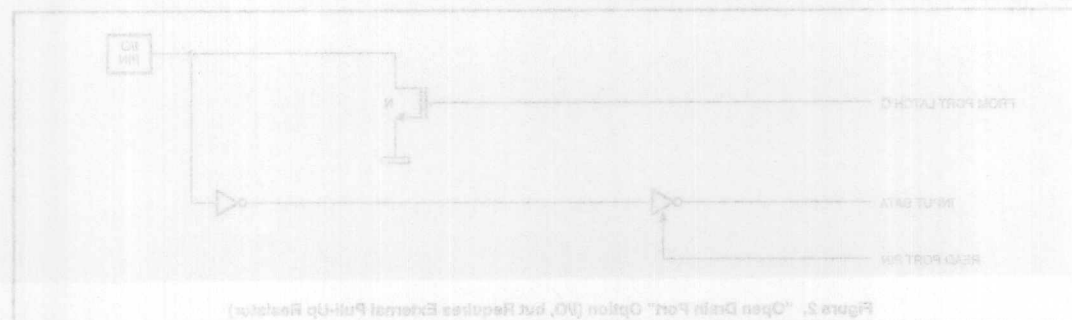


Table 2. Port Option List

I/O MASK OPTIONS SPECIFY THE DESIRED CONNECTION FOR EACH I/O LINE ON PORTS											
PORT	OPTION	PORT	OPTION	PORT	OPTION	PORT	OPTION	PORT	OPTION	PORT	OPTION
P00		P10		P20	—			P40		P50	
P01		P11		P21	—			P41		P51	
P02		P12		P22		P32		P42		P52	
P03		P13		P23		P33		P43		P53	
P04		P14		P24		P34		P44		54	
P05		P15		P25		P35		P45		—	
P06		P16		P26		P36		P46		P56	
P07		P17		P27		P37		P47		P57	
Ports option: 1: STANDARD I/O 2: OPEN-DRAIN I/O 3: PUSH-PULL OUTUT				S: HIGH AFTER POWER-ON RESET R: LOW AFTER POWER-ON RESET EXAMPLE: 3 R				P60	1 S	G	3
								P61	1 S	R	3
										FB	3





# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

## 4. IDLE AND POWER-DOWN OPERATION (for reference only)

The following paragraphs describe the general feature of P83CLxxx family and is for reference only. Since in the TV application there is no need for stop nor idle mode we will not test these two specification in the testing stage.

### 4.1 Power Control Register

Power-down and Idle modes are activated by software via the Special Function Register PCON. Its hardware address is 87H. PCON is byte addressable only.

### 4.2 Power-Down Mode

The instruction setting PCON.1 is the last executed prior to going into the Power-down mode. In Power-down mode the oscillator is stopped. The contents of the on-chip RAM and SFRs are preserved. The port pins output the values held by their respective

SFRs. The wake-up operation after power-down in this controller has two basic approaches: (section 4.3 & 5.4)

### 4.3 Wake-up from Power Down Using INT2, 3, 4, 9 (i.e., P10, P11, P12, P17)

If INT2, 3, 4, or 9 are enabled, the P83CL168 can be woken-up from power-down mode with the external interrupts. To ensure that the oscillator is stable before the controller restarts, the internal clock will remain inactive for 1536 oscillator periods. This is controlled by an on-chip delay counter.

### 4.4 Wake-up from Power Down Using RESET

To wake-up the P83CL168 the RESET pin has to be kept HIGH for a minimum of 24 cycles. The on-chip delay counter is inactive. The user has to ensure that the oscillator is stable before any operation is attempted.

### 4.5 Idle Mode

The instruction that sets PCON.0 is the last instruction executed before going into Idle mode. Once in the Idle mode, the internal clock is gated away from the CPU, but not from the interrupt, timer and Serial port functions etc. The CPU status is preserved along with the Stack pointer, Program counter, Program status Word and Accumulator. The RAM and all other registers maintain their data during Idle mode. The port pins retain the logical states they held at Idle mode activation.

### 4.6 Recover from Idle Mode

There are two methods used to terminate the Idle mode. Activation of any enabled interrupt will cause PCON to be cleared by hardware, terminating Idle mode. The interrupt is serviced, and following the instruction RETI, the next instruction to be executed will be the one following the instruction that put the device in the Idle mode.

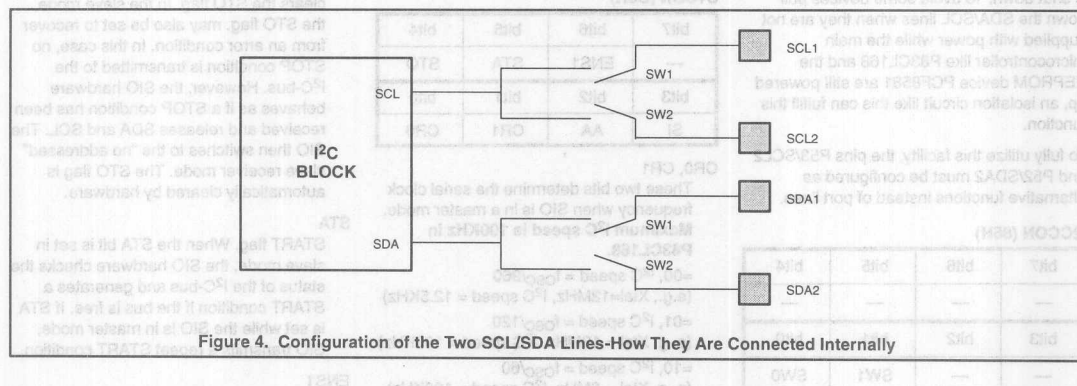


Figure 4. Configuration of the Two SCL/SDA Lines-How They Are Connected Internally



# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

## 5. I<sup>2</sup>C-BUS SERIAL I/O

### 5.1 The I<sup>2</sup>C bus and the Control Register

The serial port supports the twin line I<sup>2</sup>C-bus. The I<sup>2</sup>C-bus consists of a data line (SDA) and a clock line (SCL). These lines also function as I/O port lines P50, 51, 52 and 53 respectively. The system is unique because data transfer, clock generation, address recognition and bus control arbitration are all controlled by hardware. There is one I<sup>2</sup>C hardware only inside the P83CL168 but it is connected to two sets of SCL/SDA pins where Figure 4 shows the configuration.

An analog switch which connect or disconnect SCL1/SDA1 and/or SCL2/SDA2 pins to the internal I<sup>2</sup>C hardware is controlled by the SFR I2CCON address 86H. Application of this particular function is in the TV system, normally during the standby mode part of the power supply in the system is shut down. To avoid some devices pull down the SDA/SCL lines when they are not supplied with power while the main microcontroller like P83CL168 and the EEPROM device PCF8581 are still powered up, an isolation circuit like this can fulfill this function.

To fully utilize this facility, the pins P53/SCL2 and P52/SDA2 must be configured as alternative functions instead of port lines.

#### I2CCON (86H)

bit7	bit6	bit5	bit4
—	—	—	—
bit3	bit2	bit1	bit0
—	—	SW1	SW0

When the pins 3, 4, 5, 6 are selected as I<sup>2</sup>C data/clock lines (see chapter 9 for how to select) the following two control bits are actively control the switch connection.

- SW1 = 0, Internal SCL/SDA is disconnected from SCL1/SDA1  
 = 1, Internal SCL/SDA is connected to SCL1/SDA1
- SW2 = 0, Internal SCL/SDA is disconnected from SCL2/SDA2  
 = 1, Internal SCL/SDA is connected to SCL2/SDA2

When both SW1 and SW2 are on (=1), SCL, SCL1 and SCL2 are connected; SDA, SDA1, SDA2 are also connected.

### 5.2 Operation Modes

The I<sup>2</sup>C-bus serial I/O has complete autonomy in byte handling and operates in four modes.

- Master transmitter
- Master receiver
- Slave transmitter
- Slave receiver

These functions are controlled by the S1CON register. S1STA is the status register whose contents may also be used as a vector to various service routines. S1DAT is the data shift register and S1ADR the slave address register. Slave address recognition is performed by hardware.

For the detailed I<sup>2</sup>C specification, refer to the section 3.8 SIO1, I<sup>2</sup>C serial I/O (pp3–27 thru 3–49) of the specification 'Single chip 8-bit Microcontrollers PCB83C552/562'; Ordering information 12NC 9398 650 20011.

### 5.3 Serial Control Register S1CON

#### S1CON (D6H)

bit7	bit6	bit5	bit4
—	ENS1	STA	STO
bit3	bit2	bit1	bit0
SI	AA	CR1	CR0

#### CR0, CR1

These two bits determine the serial clock frequency when SIO is in a master mode. **Maximum I<sup>2</sup>C speed is 100KHz in P83CL168.**

- =00, I<sup>2</sup>C speed =  $f_{OSC}/960$   
 (e.g., Xtal=12MHz, I<sup>2</sup>C speed = 12.5KHz)  
 =01, I<sup>2</sup>C speed =  $f_{OSC}/120$   
 (e.g. Xtal = 12MHz, I<sup>2</sup>C speed = 100KHz)  
 =10, I<sup>2</sup>C speed =  $f_{OSC}/60$   
 (e.g. Xtal = 6MHz, I<sup>2</sup>C speed = 100KHz)  
 =11, I<sup>2</sup>C speed =  $96 * (256 - \text{Timer 1 reload range } 0-254 \text{ in mode 1})$

#### AA

Assert acknowledge bit. When the AA flag is set, an acknowledge (low level to SDA) will be returned during the acknowledge clock pulse on the SCL line when

- own slave address is received
- general call address is received (S1ADR.0=1)
- data byte received while device is programmed as master
- data byte received while device is a selected slave.

With AA=0, no acknowledge will be returned. Consequently, no interrupt is requested when the "own slave address" or general call address is received.

#### SI

SIO interrupt flag. When the S1 flag is set, an acknowledge is returned after any one of the following conditions:

- a start condition is generated in master mode.
- own slave address received during AA=1
- general call address received while S1ADR.0 and AA=1
- data byte received or transmitted in master mode (even if arbitration is lost)
- data byte received or transmitted as selected slave
- stop or start condition received as selected slave receiver or transmitter

#### STO

STOP flag. With this bit set while in master mode STOP condition is generated. When a STOP condition is detected on the bus, the SIO hardware clears the STO flag. In the slave mode, the STO flag may also be set to recover from an error condition. In this case, no STOP condition is transmitted to the I<sup>2</sup>C-bus. However, the SIO hardware behaves as if a STOP condition has been received and releases SDA and SCL. The SIO then switches to the "no addressed" slave receiver mode. The STO flag is automatically cleared by hardware.

#### STA

START flag. When the STA bit is set in slave mode, the SIO hardware checks the status of the I<sup>2</sup>C-bus and generates a START condition if the bus is free. If STA is set while the SIO is in master mode, SIO transmits a repeat START condition.

#### ENS1

When ENS1=0, the SIO is disabled. The SDA and SCL outputs are in a high-impedance state; P1.6 and P1.7 function as open drain ports. When ENS1=1, the SIO is enabled. The P1.6 and P1.7 port latches must be set to logic 1.

# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

## 5.4 Status Register S1STA

### S1STA (D9H)

bit7	bit6	bit5	bit4
SC4	SC3	SC2	SC1
bit3	bit2	bit1	bit0
SC0	0	0	0

S1STA is an 8-bit read-only special function register. S1STA.3–S1STA.7 hold a status code. S1STA.0–S1STA.2 are held LOW. The contents of S1STA may be used as a vector to a service routine. This optimizes response time of the software and consequently that of the I<sup>2</sup>C-bus.

The following is a list of the status codes:

### Abbreviations used:

SLA	7-bit slave address
R	Read bit
W	Write bit
ACK	Acknowledgement (acknowledge bit = 0)
ACKN	not Acknowledge (acknowledge bit = 1)
DATA	8-bit byte to or from the I <sup>2</sup> C-bus
MST	Master
SLV	Slave
TRX	Transmitter
REC	Receiver

### MST/TRX mode

#### S1STA value

- 08H – a START condition has been transmitted
- 10H – a repeated START condition has been transmitted
- 18H – SLA and W have been transmitted, ACK received
- 20H – SLA and W have been transmitted, ACK received
- 28H – DATA of S1DAT has been transmitted, ACK received
- 30H – DATA of S1DAT has been transmitted, ACK received
- 38H – Arbitration lost in SLA, R/W or DATA

### MST/REC mode

#### S1STA value

- 38H – Arbitration lost while returning ACK.
- 40H – SLA and R have been transmitted, ACK received.
- 48H – SLA and R have been transmitted, ACK received.
- 50H – DATA has been received, ACK returned.
- 58H – DATA has been received. ACK returned.

### SLV/REC mode

#### S1STA value

- 60H – Own SLA and W have been received, ACK returned.
- 68H – Arbitration lost in SLA, R/W as MST. Own SLA and W have been received, ACK returned.
- 70H – General CALL has been received, ACK returned.
- 78H – Arbitration lost in SLA, R/W as MST. General CALL has been received.
- 80H – Previously addressed with own SLA. DATA byte received, ACK returned.
- 88H – Previously addressed with own SLA. DATA byte received. ACK returned.
- 90H – Previously addressed with general CALL. DATA byte has been received. ACK has been returned.
- 98H – Previously addressed with general CALL. DATA byte has been received. ACK has been returned.
- A0H – A STOP condition or repeated START condition has been received while still addressed as SLV/REC or SLV/TRX.

### SLV/TRX mode

#### S1STA value

- A8H – Own SLA and R have been received. ACK returned.
- B0H – Arbitration lost in SLA, R/W as MST. Own SLA and R have been received, ACK returned.
- B8H – DATA byte has been transmitted, ACK received.
- C0H – DATA byte has been transmitted, ACK received.
- C8H – Last DATA byte has been transmitted (AA=logi 0) ACK received.

### Miscellaneous

#### S1STA value

- 00H – Bus error during MST mode or SLV mode, due to an erroneous START or STOP condition.

## 5.5 Data Shift Register S1DAT

### S1DAT (DAH)

bit7	bit6	bit5	bit4
D7	D6	D5	D4
bit3	bit2	bit1	bit0
D3	D2	D1	D0

This register contains the serial data to be transmitted or data that has just been received. Bit 7 is transmitted or received first; i.e., data is shift from left to right.

## 5.6 Slave Address S1ADR

### S1ADR (DBH)

bit7	bit6	bit5	bit4
SLA6	SLA5	SLA4	SLA3
bit3	bit2	bit1	bit0
SLA2	SLA1	SLA0	GC

GC = 0, general CALL address is not recognized;

=1, general CALL address is recognized.

SLA<6:0> own slave address

This 8-bit register may be loaded with the 7-bit slave address to which the controller will respond when programmed as slave receiver/transmitter. The LSB bit (GC) is used to determine whether the general CALL address is recognized.

For the detailed I<sup>2</sup>C specification, refer to the section 3.8 SIO1, I<sup>2</sup>C serial I/O (pp3–27 thru 3–49) of the specification 'Single chip 8-bit Microcontrollers PCB83C552/562'; Ordering information 12NC 9398 650 20011.

# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

## 6. INTERRUPT SYSTEM

External events and the real-time-driven on-chip peripherals require service by the CPU asynchronous to the execution of any particular section of code. To tie the asynchronous activities of these functions to normal program execution a multiple-source, two-priority-level, nested interrupt system is provided. The P83CL168 acknowledges interrupt requests from thirteen sources as shown in Table 3.

Each interrupt vectors to a separate location in program memory for its service routine. Each source can be individually enabled or disabled by corresponding bits in the interrupt Enable Registers (IE, IEO). The priority level is selected via the interrupt priority register (IPO, IP1). All enabled sources can be globally disabled or enabled.

## 6.1 External Interrupts INT2, 3, 4 and INT9 and Interrupts 5, 6, 7, 8 (used by derivative functions)

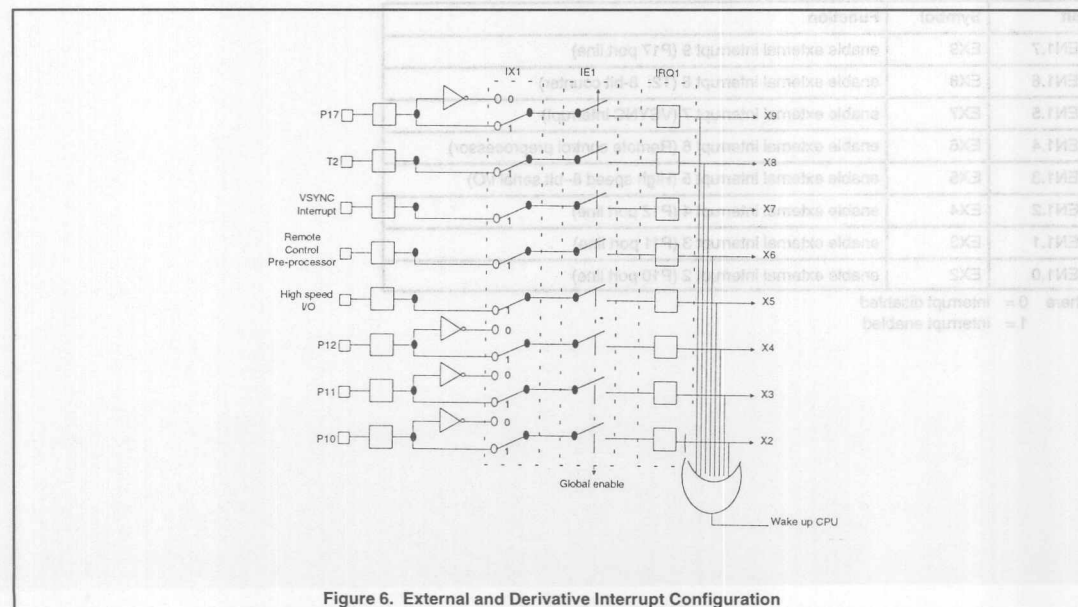
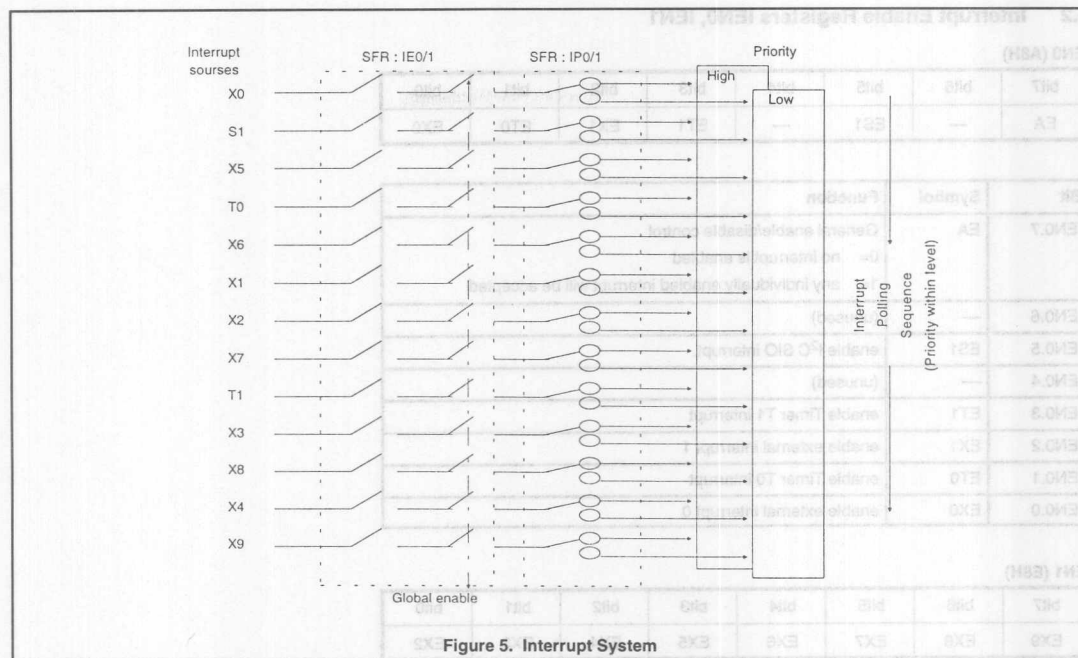
Port 1 lines serve an alternative purpose as four additional interrupts INT2, 3, 4 and 9 (P10, P11, P12 and P17). The Interrupt INT5, 6, 7 and 8 which were connected to port P13, 14, 15 and 16 in the standard P83CL410 are no longer connected to these pins. Instead, they are used separately by the derivative functional blocks as follows:

- X5 High speed 8-bit serial I/O 0053H
- X6 Remote Control preprocessor 005BH
- X7 VSYNC interrupt 0063H
- X8 T2 (AC frequency counter) 006BH

Using the 1X1 register, each pin may be initialized to either active HIGH or LOW. (INT2, 3, 4 & 9 only, 5, 6, 7, 8 are fixed to active high because these interrupt are from other derivative functions.) IRQ1 is the interrupt request flag register. Each flag, if the interrupt is enabled, will be set on interrupt request but it must be cleared by software, i.e., via the interrupt software or when the interrupt is disabled.

Table 3.

PRIORITY (within level)	SOURCE	VECTOR (address in HEX decimal)
PX0 (Highest)	External Interrupt 0 (INT0N)	0003H
S1	I <sup>2</sup> C	002BH
PX5	High speed 8-bit serial I/O	0053H
T0	Timer 0	000BH
PX6	Remote Control preprocessor	005BH
PX1	External Interrupt 1 (INT1N)	0013H
PX2	P10 (port line 10)	003BH
PX7	VSYNC interrupt	0063H
T1	Timer 1	001BH
PX3	P11 (port line 11)	0043H
PX8	T2 (AC frequency counter)	006BH
PX4	P12 (port line 12)	004BH
PX9 (Lowest)	P17 (port line 17)	0073H



# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

## 6.2 Interrupt Enable Registers IEN0, IEN1

### IEN0 (A8H)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
EA	—	ES1	—	ET1	EX1	ET0	EX0

Bit	Symbol	Function
IEN0.7	EA	General enable/disable control 0= no interrupt is enabled 1= any individually enabled interrupt will be accepted
IEN0.6	—	(unused)
IEN0.5	ES1	enable I <sup>2</sup> C SIO interrupt.
IEN0.4	—	(unused)
IEN0.3	ET1	enable Timer T1 interrupt
IEN0.2	EX1	enable external interrupt 1
IEN0.1	ET0	enable Timer T0 interrupt
IEN0.0	EX0	enable external interrupt 0

### IEN1 (E8H)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
EX9	EX8	EX7	EX6	EX5	EX4	EX3	EX2

Bit	Symbol	Function
IEN1.7	EX9	enable external interrupt 9 (P17 port line)
IEN1.6	EX8	enable external interrupt 8 (T2: 8-bit counter)
IEN1.5	EX7	enable external interrupt 7 (VSYNC interrupt)
IEN1.4	EX6	enable external interrupt 6 (Remote control preprocessor)
IEN1.3	EX5	enable external interrupt 5 (High speed 8-bit serial I/O)
IEN1.2	EX4	enable external interrupt 4 (P12 port line)
IEN1.1	EX3	enable external interrupt 3 (P11 port line)
IEN1.0	EX2	enable external interrupt 2 (P10 port line)

where 0 = interrupt disabled  
1 = interrupt enabled



# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

## 6.3 Interrupt Priority Registers IP0, IP1

IP0 (B8H)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	PS1	—	PT1	PX1	PT0	PX0

Bit	Symbol	Function
IP.7	—	(unused)
IP.6	—	(unused)
IP.5	PS1	I <sup>2</sup> C SIO interrupt priority level
IP.4	—	(unused)
IP.3	PT1	Timer 1 interrupt priority level
IP.2	PX1	External interrupt priority level.
IP.1	PT0	Timer 0 interrupt priority level.
IP.0	PX0	External interrupt 0 priority level.

IP1 (F8H)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
PX9	PX8	PX7	PX6	PX5	PX4	PX3	PX2

Bit	Symbol	Function
IP1.7	PX9	external interrupt 9 priority level (P17 port line)
IP1.6	PX8	external interrupt 8 priority level (T2 : 8-bit counter)
IP1.5	PX7	external interrupt 7 priority level (VSYNC interrupt)
IP1.4	PX6	external interrupt 6 priority level (Remote control preprocessor)
IP1.3	PX5	external interrupt 5 priority level (High speed 8-bit serial I/O)
IP1.2	PX4	external interrupt 4 priority level (P12 port line)
IP1.1	PX3	external interrupt 3 priority level (P11 port line)
IP1.0	PX2	external interrupt 2 priority level (P10 port line)

Interrupt priority is as follows: 0—low priority;  
1—high priority



## 6.4 Interrupt Polarity Register IX1

IX1 (E9H)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
IL9	1	1	1	1	IL4	IL3	IL2

Writing either a "1" or "0" to an IX1 register bit sets the polarity level of the corresponding external interrupt to active HIGH (=1) or LOW (=0) respectively. But IX1<6:3> is fixed to 1 (active high).

Bit	Symbol	Function
IX1.7	IL9	external interrupt 9 polarity level (P17 port line)
IX1.6	IL8	external interrupt 8 polarity level (T2: 8-bit counter)
IX1.5	IL7	external interrupt 7 polarity level (VSYNC interrupt)
IX1.4	IL6	external interrupt 6 polarity level (Remote control preprocessor)
IX1.3	IL5	external interrupt 5 polarity level (High speed 8-bit serial I/O)
IX1.2	IL4	external interrupt 4 polarity level (P12 port line)
IX1.1	IL3	external interrupt 3 polarity level (P11 port line)
IX1.0	IL2	external interrupt 2 polarity level (P10 port line)

## 6.5 Interrupt Request Flag Register IRQ1

IRQ1 (C0H)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
IQ9	IQ8	IQ7	IQ6	IQ5	IQ4	IQ3	IQ2

Bit 7, 2, 1, 0 are set to one when the corresponding bit in IEN1 SFR is enabled and its corresponding port line 17, 12, 11, 10 is high (if polarity IX1.y, y=7, 2, 1, 0, is set to high). A pending interrupt is generated to CPU. If IEN1.7, 2, 1 or 0 is 0 (disabled) then IRQ1.7, 2, 1, 0 will never set to 1 even the external port line condition is met. Reset of IRQ1.7, 2, 1, 0 can be done by software write "0" only.

But for IRQ1.6, 5, 4, 3, these four bits serve not only as a pending interrupt request bits like IRQ1.7, 2, 1, 0 but also as the interrupt status bits. This means even if IEN1.6, 5, 4, 3 is NOT enabled, the IRQ1.6, 5, 4, 3 can still be set to one if the interrupt condition is met within the corresponding derivative functions.

That is, for example, if the interrupt condition within Counter T2 is met (i.e., overflow), then:

- If IEN0.7 = X, IEN1.6 = 0 → then IRQ1.6 = 1, no pending interrupt to CPU.
- If IEN0.7 = 0, IEN1.6 = 1 → then IRQ1.6 = 1, interrupt to CPU is pending.
- If IEN0.7 = 1, IEN1.6 = 1 → then IRQ1.6 = 1, interrupt will be served as soon as
  - CPU finishes current instruction, if not in the interrupt service routine.
  - or the current interrupt service routine is interrupted if the T2 has higher interrupt priority
  - or this T2 interrupt becomes pending, waiting until the current higher priority level interrupt is served.

Bit	Symbol	Function
IRQ1.7	IQ9	external interrupt 9 request flag (P17 port line)
IRQ1.6	IQ8	external interrupt 8 request flag (T2: 8-bit counter)
IRQ1.5	IQ7	external interrupt 7 request flag (VSYNC interrupt, see chapter 13)
IRQ1.4	IQ6	external interrupt 6 request flag (Remote control preprocessor, see chapter 14)
IRQ1.3	IQ5	external interrupt 5 request flag (High speed 8-bit serial I/O, see chapter 15)
IRQ1.2	IQ4	external interrupt 4 request flag (P12 port line)
IRQ1.1	IQ3	external interrupt 3 request flag (P11 port line)
IRQ1.0	IQ2	external interrupt 2 request flag (P10 port line)

# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

## 6.6 Interrupt Priority

Each interrupt source can be set to either high or low priority. If both priorities are requested simultaneously, the controller will branch to the high priority vector.

A low priority interrupt can only be interrupted by a high priority interrupt. A high priority interrupt routine cannot be interrupted.

## 6.7 Related Registers

The following registers are used in conjunction with the interrupt system:

Register	Function	SFR Address
1X1	interrupt polarity register	E9H
IRQ1	interrupt request flag register	C0H
IEN0	interrupt enable register	A8H
IEN1	interrupt enable register (INT2-INT9)	E8H
IP0	interrupt priority register	B8H
IP1	interrupt priority register (INT2-INT9)	F8H

## 7. OSCILLATOR CIRCUITRY

The on-chip oscillator circuitry of the P83CL168 is a single-stage inverting amplifier biased by an internal feedback resistor (Figure 7). For operation as a standard quartz oscillator, no external components are needed. When using external ceramic resonators, different configuration is supported (see Figure 8 and oscillator options).

### 7.1 Oscillator Options

Table 4 lists oscillator transconductance options as provided for optimum on-chip oscillator performance. Please state option when ordering.

Table 4. Oscillator Options Specify the Desired Oscillator Frequency

Option	Typ. Gm @ 5V	f <sub>osc</sub> for Quartz	f <sub>osc</sub> for Ceramic
HIGH (GmH)	4.5mS/V	10 - 16MHz	3 - 16MHz
MED (GmM)	1.8mS/V	4 - 12MHz	1 - 8MHz

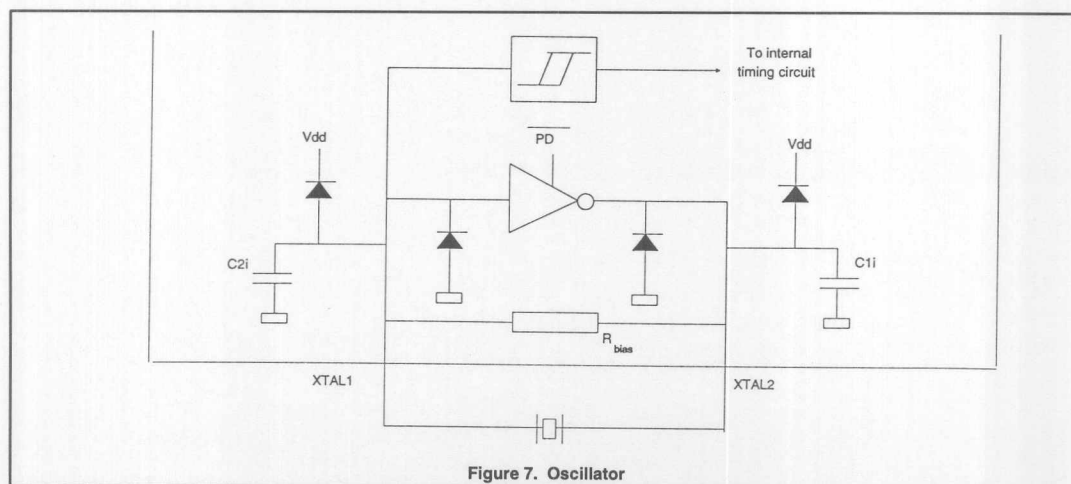


Figure 7. Oscillator

Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

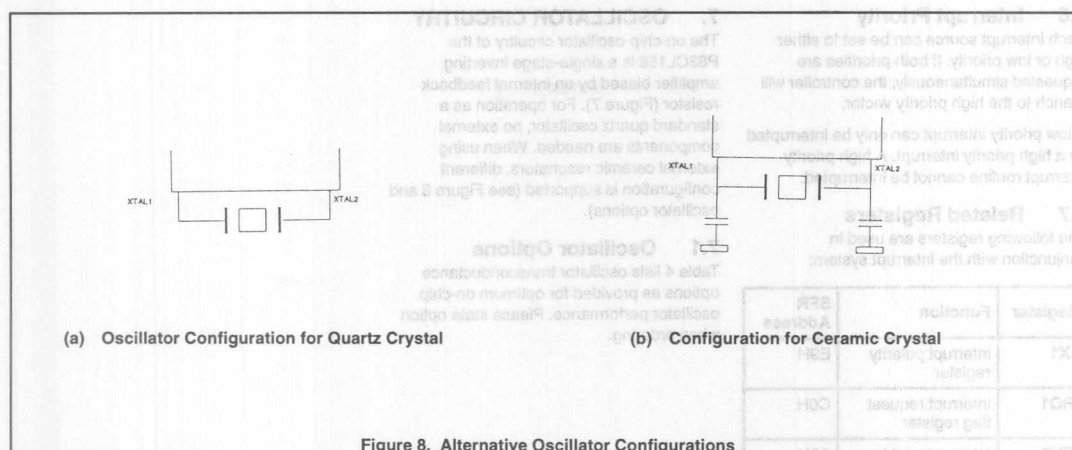


Figure 8. Alternative Oscillator Configurations

Table 4.

Transconductance (Gm) Mask Options Specify the Desired Oscillator Frequency

	Option	Typ. Gm @ 5V	f <sub>osc</sub> for Quartz	f <sub>osc</sub> for Ceramic Resonator	NOTE:
<input type="checkbox"/>	MED (GmM)	1.6mA/V	4 – 12MHz	1 – 6MHz	The frequency used in P83CL168 spec. is 4 to 12MHz.
<input type="checkbox"/>	HIGH (GmH)	4.5mA/V	10 – 16MHz	3 – 16MHz	

# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

## 8. RESET CIRCUITRY AND SFR MAP

To initialize the P83CL168 a reset is performed by either of two methods:

- via the RST pin
- via a power-on-reset

It leaves the internal registers as follows:

SFR addr	Register	Content
E0H	ACC	0000 0000
F0H	B	0000 0000
82H	DPL	0000 0000
83H	DPH	0000 0000
A8H	IE0	0000 0000
E8H	IE1	0000 0000
B8H	IP0	xx00 0000
F8H	IP1	0000 0000
E9H	IX1	0000 0000
C0H	IRQ1	0000 0000
D0H	PSW	0000 0000
87H	PCON	0xxx 0000
80H	P0	1111 1111
90H	P1	1111 1111
A0H	P2	1111 1111
B0H	P3	1111 1111
B1H	P4	1111 1111
B2H	P5	1111 1111
B3H	P6	xxxx xx11
DBH	S1ADR	0000 0000
D8H	S1CON	X000 0000
DAH	S1DAT	0000 0000
D9H	S1SAT	1111 1000
81H	SP	0000 0111
88H	TCON	0000 0000
8CH	TH0	0000 0000

8DH	TH1	0000 0000
8AH	TL0	0000 0000
8BH	TL1	0000 0000
89H	TMOD	0000 0000
86H	I2CCON	xxxx xx00
98H	OSAT	xxx1 1111
99H	OSDT	1111 1111
9AH	OSAD	0000 0000
CBH	RCMV	0000 0000
CAH	RCM	xx00 0000
C1H	OSCON	xx01 1100
C2H	OSORGV	xx11 1111
C3H	OSORGH	xx11 1111
C4H	OSPLL	xx01 0000
C5H	OSSTART	0000 0000
C6H	HIOD	xxxx xxxx
C7H	HIOCON	xxx0 0000
C8H	AFCON	x000 000x
C9H	VINT	xxxx xxxx
CCH	P1SEL	0000 0000
CDH	P4SEL	0000 0000
CEH	P5SEL	0000 0000
CFH	OSCON2	xx00 000x
D2H	VSTL	x000 0000
D3H	VSTH	x000 0000
E4H	PWM0	x000 0000
E5H	PWM1	x000 0000
E6H	PWM2	x000 0000
E7H	PWM3	x000 0000
ECH	PWM4	xx00 0000
EDH	PWM5	xx00 0000
EEH	PWM6	xx00 0000
EFH	PWM7	xx00 0000
8EH	T2	0000 0000

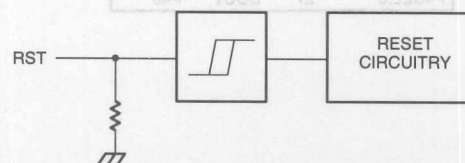
The reset state of the port pins is mask-programmable and can therefore be defined by the customer.

The output of the Schmitt trigger is sampled by the reset circuitry every machine cycle.

A reset is accomplished by holding the RST pin HIGH for at least two machine cycles (24 oscillator periods), while the oscillator is running. The CPU responds by generating an internal reset. Port pins adopt their reset state immediately after RST goes HIGH.

The external reset is asynchronous to the internal clock. The RST pin is sampled during State 5 phase 2 of every machine cycle. After a HIGH is detected at the RST pin, an internal reset is repeated every cycle until RST goes LOW.

The internal RAM is not affected by reset. When  $V_{DD}$  is returned on the RAM contents remain unchanged.



The rest input to the P83CL168 is RST pin. A Schmitt trigger qualifies the input for noise rejection.

Figure 9. Reset Configuration at RST Pin

# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

## 9. PORT/ALTERNATIVE FUNCTION SELECTION

The port lines 1, 3, 4 and 5 are shared with the derivative functions' input or output pin. To configure these pins to either port lines or alternative functions, the following SFR are required.

### 9.1 Port 1 Selection – P1SEL

#### P1SEL (CCH)

bit7	bit6	bit5	bit4
PWM7E	PWM6E	PWM5E	PWM4E
bit3	bit2	bit1	bit0
PWM3E	PWM2E	PWM1E	PWM0E

Port 1 is shared with the 6 or 7-bit PWM's (pins 8, 9, 10, 21, 22, 23, 24, 25). When the corresponding bit  $i$  ( $i=7-0$ ) in the P1SEL is programmed to 1, the pin functions as the output of the PWM $i$ . If that bit is '0' then port line 1 $i$  is connected to that pin.

Abbreviations used in the following table,

Bit: bit  $i$  ( $i=7-0$ ) of the register

Pin: pin number which contains the multiplexed I/O and alternative function

=1: when that bit =1, the pin  $xx$  becomes the alternative function indicated.

=0: when that bit = 0, the pin  $xx$  becomes the port line  $yy$ .

Bit	pin	= 1	= 0
P1SEL.7	44	PWM7	P17
P1SEL.6	43	PWM6	P16
P1SEL.5	42	PWM5	P15
P1SEL.4	41	PWM4	P14
P1SEL.3	40	PWM3	P13
P1SEL.2	39	PWM2	P12
P1SEL.1	38	PWM1	P11
P1SEL.0	37	PWM0	P10

### 9.2 Port 3 Selection

Part of the port 3 lines are multiplexed with the input pins of the external interrupt INT0N and INT1N and the timer/counter input pins T0 and T1. To configure these pins into the alternative input functions, the corresponding bit in the port latch should be programmed to 1. No special control required for this selection. Port 31 is no longer available.

### 9.3 Port 4 Selection – P4SEL

#### P4SEL (CDH)

bit7	bit6	bit5	bit4
AFC3E	AFC2E	AFC1E	AFC0E
bit3	bit2	bit1	bit0
TDACE	DINE	CLKE	DOUTE

An '1' in the corresponding bit enable the pin as the alternative function.

A '0' makes the pin as a general I/O port line.

AFC3E, AFC2E, AFC1E, AFC0E are to control the pins 34, 33, 32 and 31 as either port line 47, 46, 45 and 44 or the Automatic Frequency Control inputs 3, 2, 1, or 0.

TDACE is to control the pin 30 as 14-bit PWM output or the I/O port 43.

DINE, CLKE, DOUTE are to control the pins 29, 28, 27 as the high speed 8-bit serial I/O function or port line 42, 41 and 40.

Bit	pin	= 1	= 0
P4SEL.7	34	AFC3	P47
P4SEL.6	33	AFC2	P46
P4SEL.5	32	AFC1	P45
P4SEL.4	31	AFC0	P44
P4SEL.3	30	TDAC	P43
P4SEL.2	29	DIN	P42
P4SEL.1	28	CLK	P41
P4SEL.0	27	DOUT	P40

### 9.4 Port 5 Selection – P5SEL

#### P5SEL (CEH)

bit7	bit6	bit5	bit4
RMCE	T2E	—	OSDBE
bit3	bit2	bit1	bit0
SCL2E	SDA2E	SCL1E	SDA1E

An '1' in the corresponding bit enable the pin as the alternative function.

A '0' makes the pin as a general I/O port line.

Port 55 is no longer available.

Bit	pin	= 1	= 0
P5SEL.7	26	RMC	P57
P5SEL.6	25	T2	P56
P5SEL.4	58	B	P54
P5SEL.3	3	SCL2	P53
P5SEL.2	4	SCL1	P52
P5SEL.1	5	SDA2	P51
P1SEL.0	6	SDA1	P50

# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

## 10. ANALOG CONTROL:

### 6-BIT/7-BIT PWMs

The P83CL168-VST has eight PWM DAC outputs for analog controls of, e.g., volume, balance, brightness and saturation. These PWM outputs generate pulse patterns with a repetition rate of  $F_{\text{pwm}}/64$  or  $F_{\text{pwm}}/128$ . The analog value is determined by the ratio of the HIGH-time and the repetition time. A DC voltage proportional to the PWM control setting is obtained by means of an external integration network (low pass filter). PWM0-3 is 7 bit while PWM4-7 is 6 bit.

The polarity of the PWMn output is fixed to active high. For those people who are familiar with Philips PCA84C846 TV microcontroller architecture, the active high output means P6LVL and P7LVL bit of derivative register 23H are programmed to '0'. Refer to PCA84C846 specification for details.

### 10.1 SFRs for PWMs Control

The following SFR's are the data registers of the 7-bit PWM's and 6-bit PWM's.

#### PWM0 (E4H)

bit7	bit6	bit5	bit4
—	data6	data5	data4
bit3	bit2	bit1	bit0
data3	data2	data1	data0

#### PWM1 (E5H)

bit7	bit6	bit5	bit4
—	data6	data5	data4
bit3	bit2	bit1	bit0
data3	data2	data1	data0

#### PWM2 (E6H)

bit7	bit6	bit5	bit4
—	data6	data5	data4
bit3	bit2	bit1	bit0
data3	data2	data1	data0

#### PWM3 (E7H)

bit7	bit6	bit5	bit4
—	data6	data5	data4
bit3	bit2	bit1	bit0
data3	data2	data1	data0

#### PWM4 (E8H)

bit7	bit6	bit5	bit4
—	—	data5	data4
bit3	bit2	bit1	bit0
data3	data2	data1	data0

#### PWM5 (EDH)

bit7	bit6	bit5	bit4
—	—	data5	data4
bit3	bit2	bit1	bit0
data3	data2	data1	data0

#### PWM6 (EEH)

bit7	bit6	bit5	bit4
—	—	data5	data4
bit3	bit2	bit1	bit0
data3	data2	data1	data0

#### PWM7 (EFH)

bit7	bit6	bit5	bit4
—	—	data5	data4
bit3	bit2	bit1	bit0
data3	data2	data1	data0

The high time of a PWMn output is:

$$T_{\text{high}} = [\text{PWMn}] \cdot t_0$$

where [PWMn] is the "contents of PWMn data latch" and

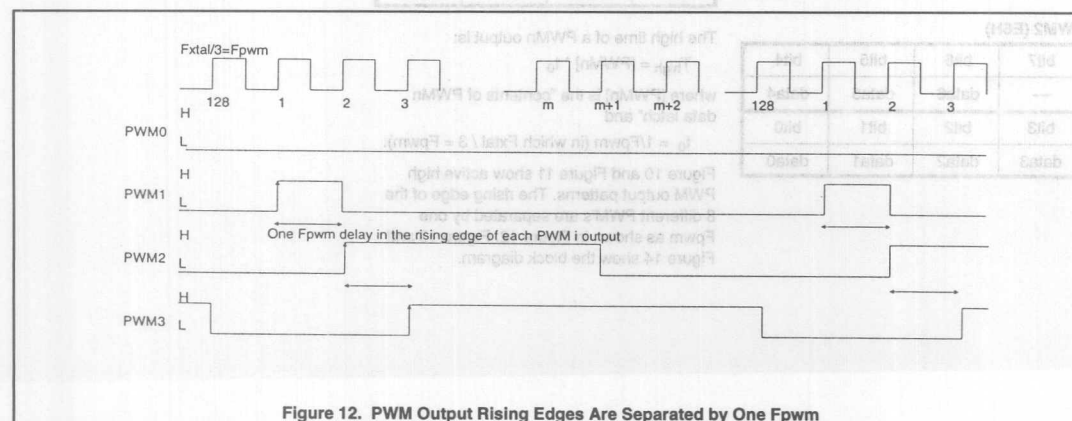
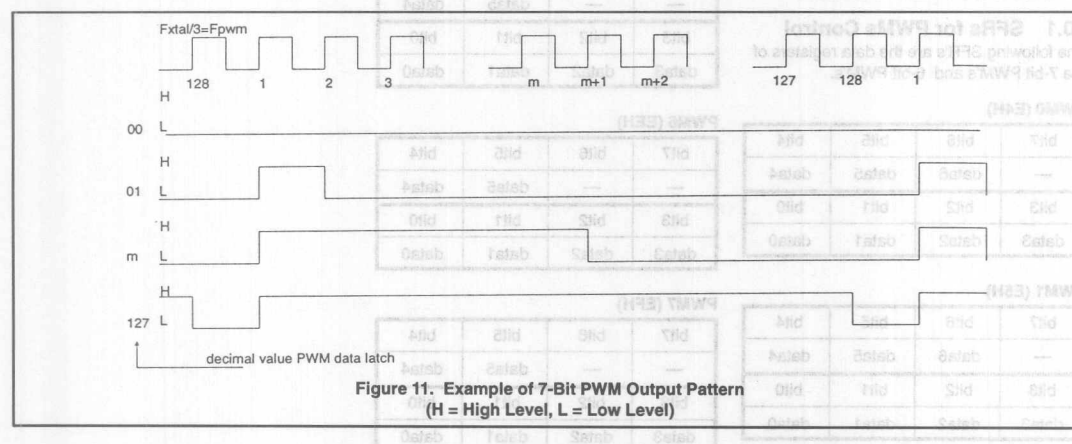
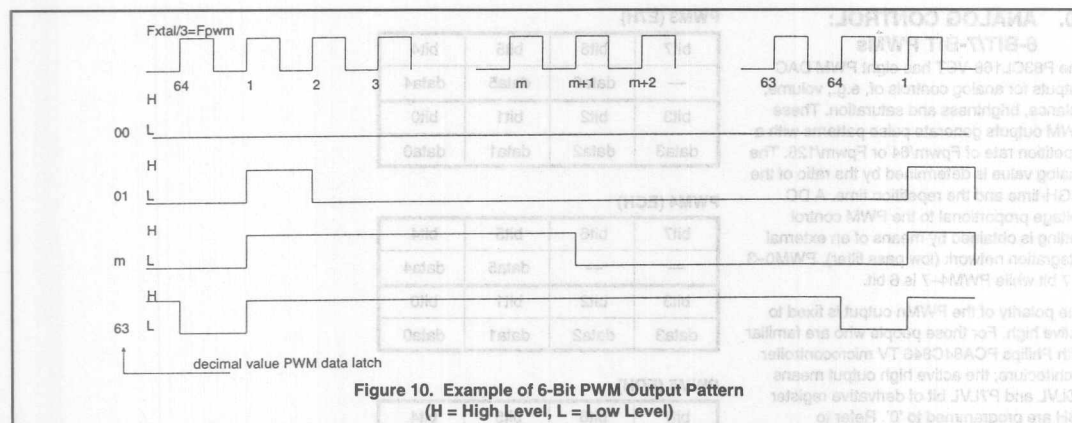
$$t_0 = 1/F_{\text{pwm}} \text{ (in which } F_{\text{xtal}} / 3 = F_{\text{pwm}} \text{).}$$

Figure 10 and Figure 11 show active high PWM output patterns. The rising edge of the 8 different PWM's are separated by one  $F_{\text{pwm}}$  as shown in Figure 12. Figure 13 and Figure 14 show the block diagram.



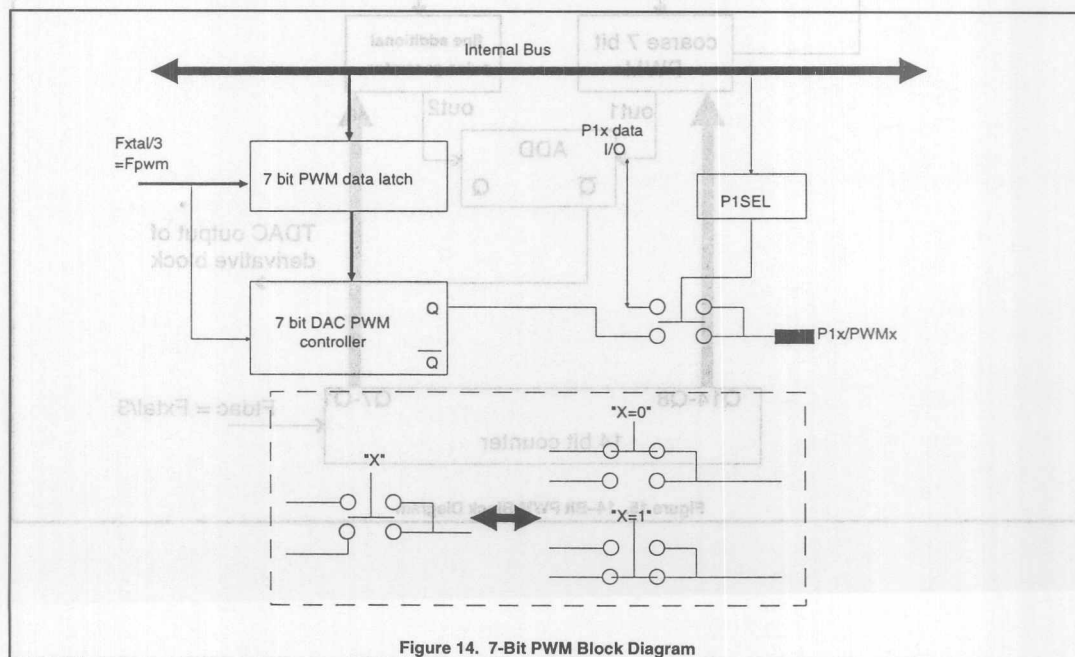
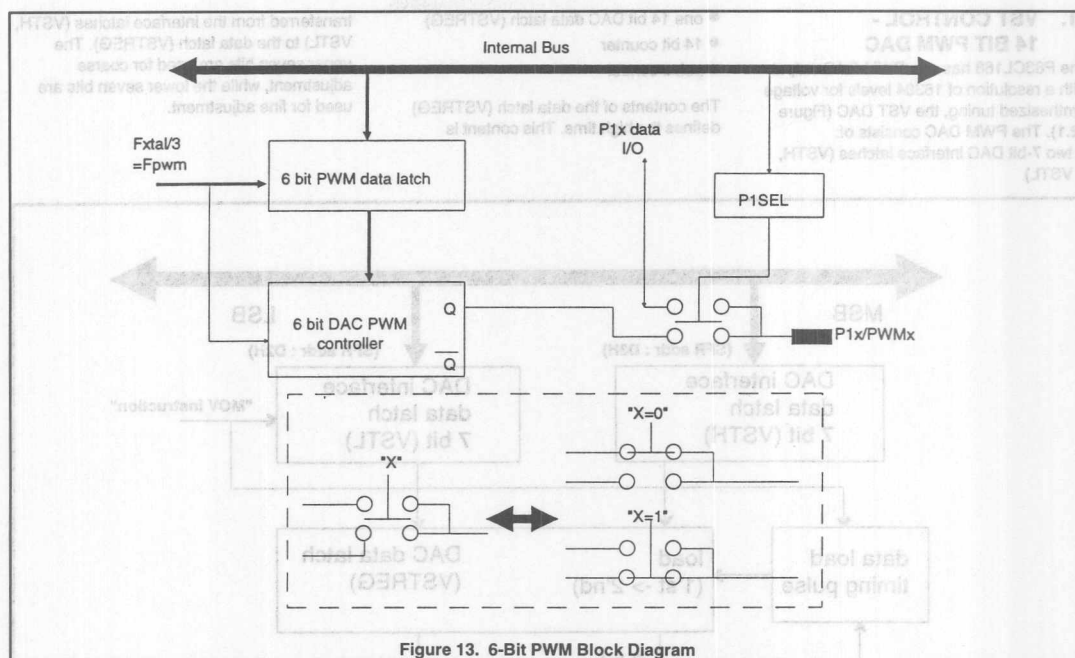
# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267



# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267



# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

## 11. VST CONTROL - 14 BIT PWM DAC

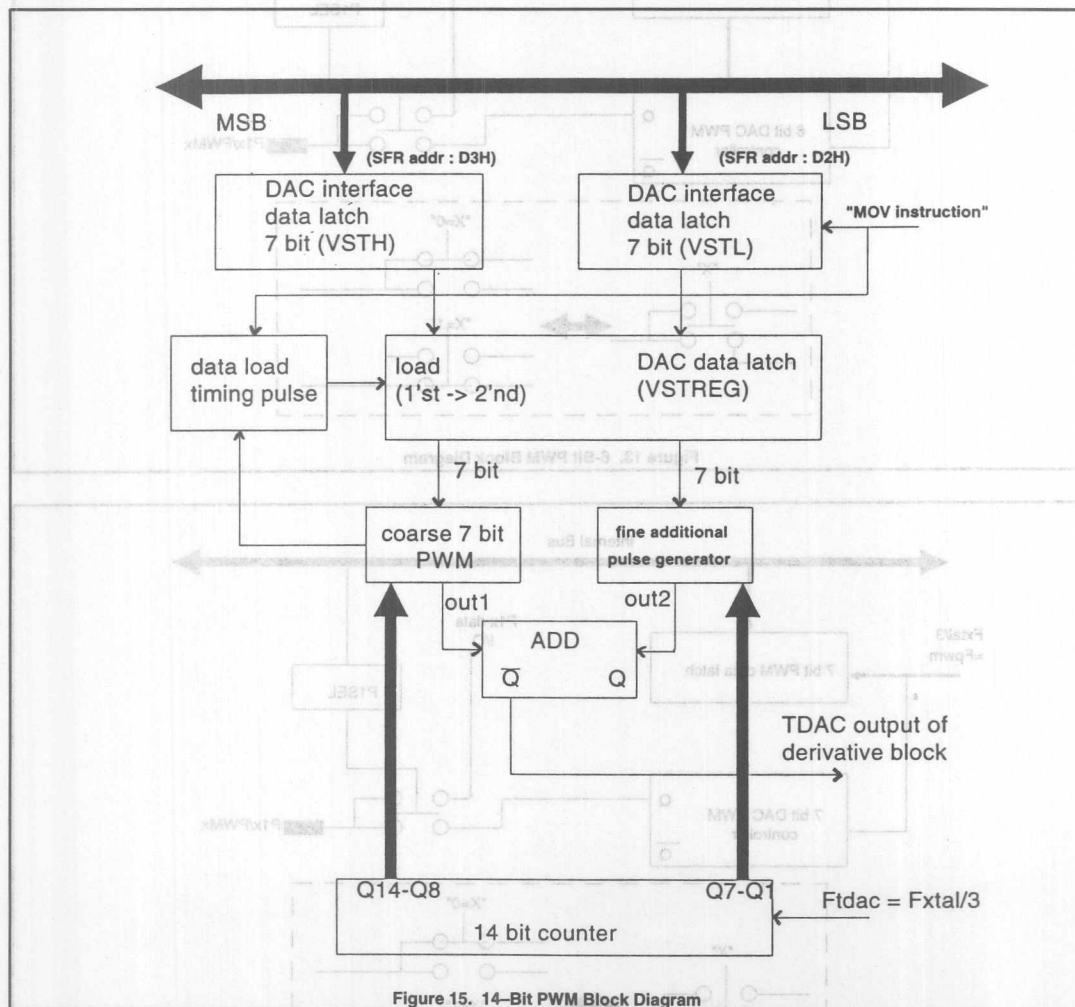
The P83CL168 has one PWM DAC output with a resolution of 16384 levels for voltage synthesized tuning, the VST DAC (Figure 12.1). The PWM DAC consists of:

- two 7-bit DAC interface latches (VSTH, VSTL)

- one 14 bit DAC data latch (VSTREG)
- 14 bit counter
- pulse control

The contents of the data latch (VSTREG) defines the high time. This content is

transferred from the interface latches (VSTH, VSTL) to the data latch (VSTREG). The upper seven bits are used for coarse adjustment, while the lower seven bits are used for fine adjustment.



# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

## 11.1 Repetition Time

upper 7 bits (VSTH):

$$F_{th} = F_{tdac} / 128$$

lower 7 bits (VSTL):

$$F_{tl} = F_{th} / 128 = F_{tdac} / 16384$$

The data from "VSTH", "VSTL" is latched into "VSTREG" at the beginning of the first  $T_{sub}$  after "VSTL" is written. So, in order to be sure that the correct data is latched into "VSTREG", "VSTH" must contain the right value before "VSTL" is written. (Fig 16, Note: in case 1 and case 2, a new value for VSTH, VSTL is latched into VSTREG. In

case 3, VSTL, together with an old value of VSTH are latched into VSTREG):

After VSTH, VSTL is latched into VSTREG, it takes one  $T_{sub}$  to generate the appropriate pulse pattern. So to ensure correct D/A converting, one should wait for two  $T_{sub}$  before changing VST(H)(L) again.

*Different from the 6-bit/7-bit PWM's active high, the output of the 14-bit PWM is active low; i.e., the high period of the output pulse is the value stored in the VSTH/VSTL special function register.*

## 11.2 Coarse Adjustment (VSTH)

VSTH (D3H)

bit7	bit6	bit5	bit4
—	VSTD	VSTC	VSTB
bit3	bit2	bit1	bit0
VSTA	VST9	VST8	VST7

Coarse adjustment (Figure 17) is done in periods of  $T_{sub}$ . During 1  $T_{sub}$ , the coarse adjustment output (Out1 of Figure 15) is high while  $t_g \leq [VSTH+1] * t_0$ . The coarse adjustment output is high while  $[VSTH] * t_0 < t_g < 128 * t_0$ . Where  $t_g$  is the time within  $T_{sub}$ .

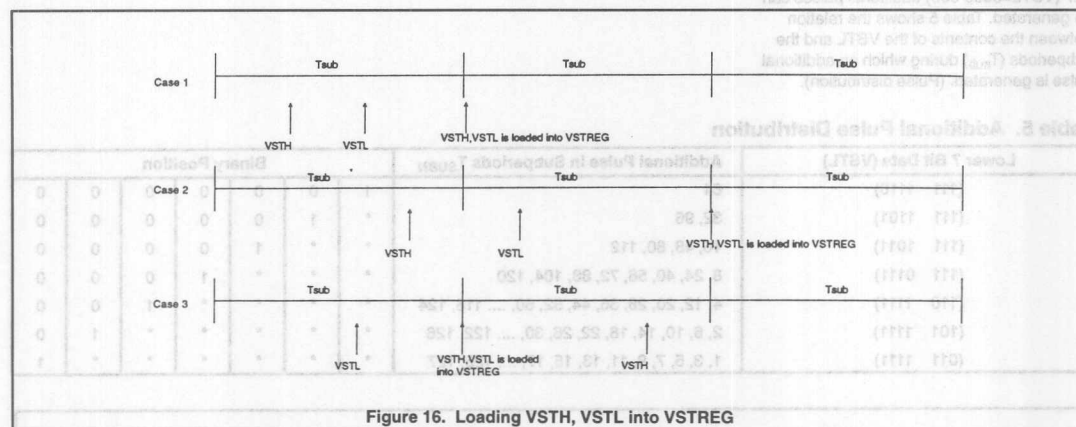


Figure 16. Loading VSTH, VSTL into VSTREG

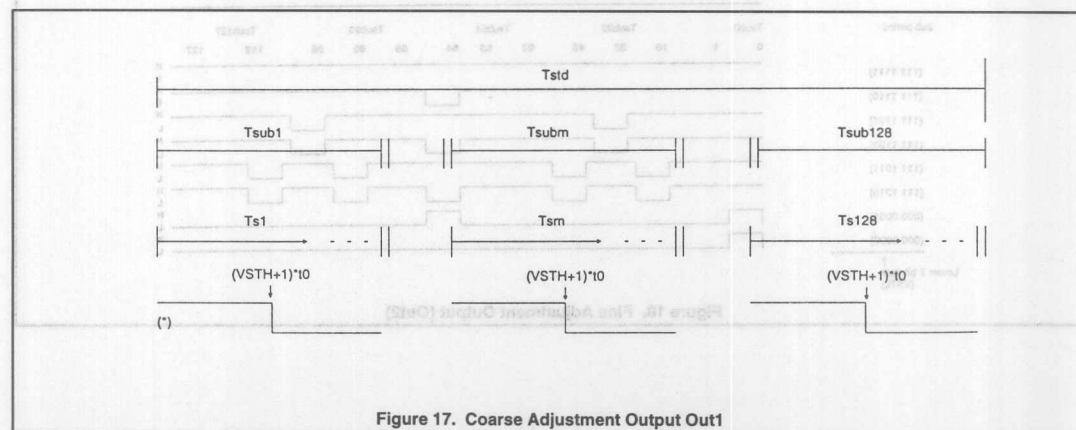


Figure 17. Coarse Adjustment Output Out1

# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

## P83CL168/167/268/267

### 11.3 Fine adjustment (VSTL)

#### VSTL (D2H)

bit7	bit6	bit5	bit4
VST6	VST6	VST5	VST4
bit3	bit2	bit1	bit0
VST3	VST2	VST1	VST0

Fine adjustment is done with *additional* pulses, generated in particular subperiods. The contents of VSTL indicates in which periods an additional pulse is generated.

During Tstd, 0 (VSTL=1111 111) to 127 (VSTL=0000 000) additional pulses can be generated. Table 5 shows the relation between the contents of the VSTL and the subperiods ( $T_{sub}$ ) during which an additional pulse is generated. (Pulse distribution).

If, for example VSTH="0011101" VSTL="1111101", then TDAC looks like in Figure 20. The additional pulses are generated in subperiods 32 and 96.

Table 5. Additional Pulse Distribution

Lower 7 Bit Data (VSTL)	Additional Pulse in Subperiods $T_{SUBN}$	Binary Position						
(111 1110)	64	1	0	0	0	0	0	0
(111 1101)	32, 96	*	1	0	0	0	0	0
(111 1011)	16, 48, 80, 112	*	*	1	0	0	0	0
(111 0111)	8, 24, 40, 56, 72, 88, 104, 120	*	*	*	1	0	0	0
(110 1111)	4, 12, 20, 28, 36, 44, 52, 60, .... 116, 124	*	*	*	*	1	0	0
(101 1111)	2, 6, 10, 14, 18, 22, 26, 30, .... 122, 126	*	*	*	*	*	1	0
(011 1111)	1, 3, 5, 7, 9, 11, 13, 15, 17, .... 125, 127	*	*	*	*	*	*	1

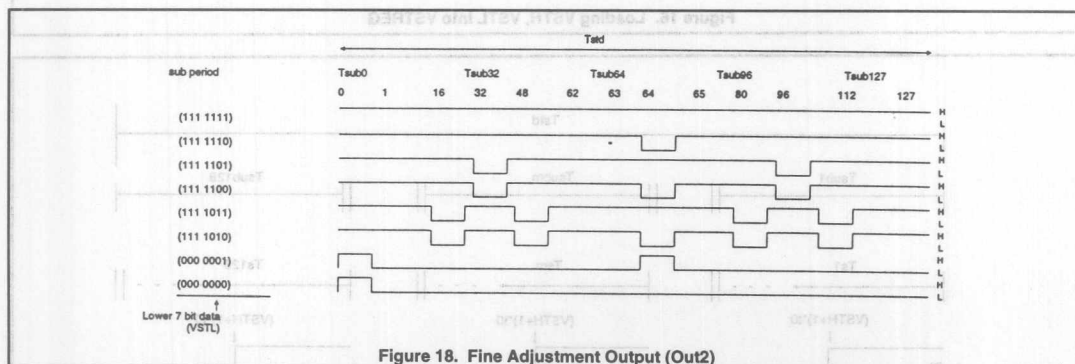
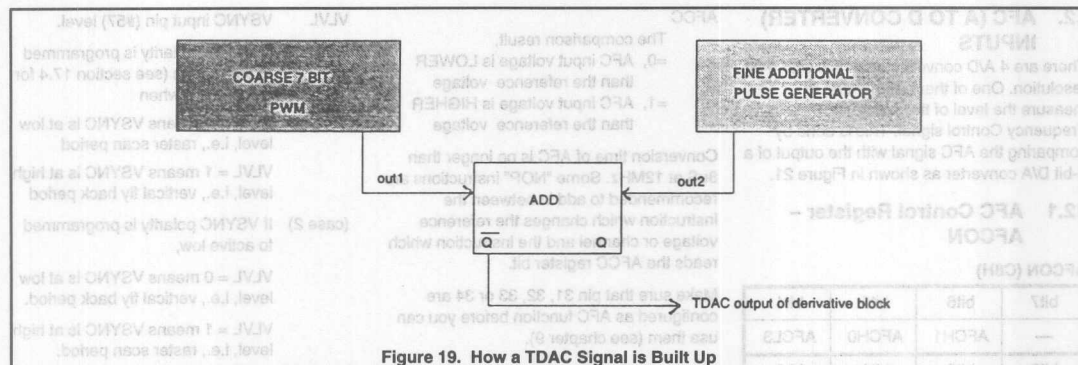


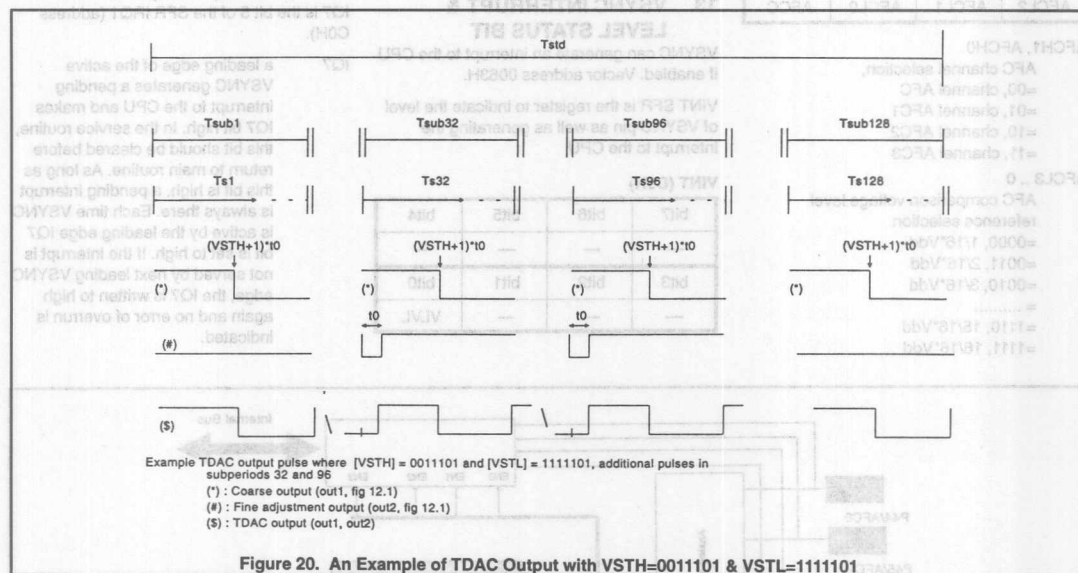
Figure 18. Fine Adjustment Output (Out2)

**Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function**

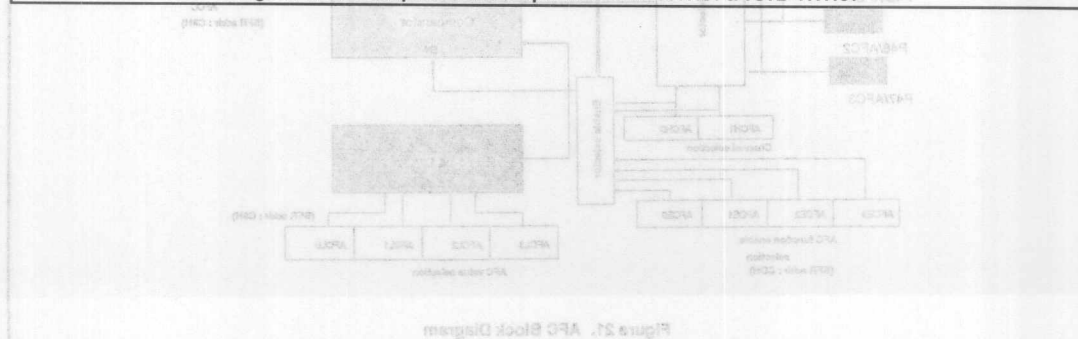
P83CL168/167/268/267



**Figure 19. How a TDAC Signal is Built Up**



**Figure 20. An Example of TDAC Output with VSTH=0011101 & VSTL=1111101**





# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

## 12. AFC (A TO D CONVERTER) INPUTS

There are 4 A/D converter channels with 4-bit resolution. One of them is intended to measure the level of the Automatic Frequency Control signal. This is done by comparing the AFC signal with the output of a 4-bit D/A converter as shown in Figure 21.

### 12.1 AFC Control Register – AFCON

AFCON (C8H)

bit7	bit6	bit5	bit4
—	AFCH1	AFCH0	AFCL3
bit3	bit2	bit1	bit0
AFCL2	AFCL1	AFCL0	AFCC

AFCH1, AFCH0

AFC channel selection,  
=00, channel AFC  
=01, channel AFC1  
=10, channel AFC2  
=11, channel AFC3

AFCL3 .. 0

AFC comparison voltage level  
reference selection  
=0000,  $1/16 \cdot V_{DD}$   
=0011,  $2/16 \cdot V_{DD}$   
=0010,  $3/16 \cdot V_{DD}$   
= .....  
=1110,  $15/16 \cdot V_{DD}$   
=1111,  $16/16 \cdot V_{DD}$

AFCC

The comparison result,  
=0, AFC input voltage is LOWER  
than the reference voltage  
=1, AFC input voltage is HIGHER  
than the reference voltage

Conversion time of AFC is no longer than 8 $\mu$ S at 12MHz. Some "NOP" instructions are recommended to add in between the instruction which changes the reference voltage or channel and the instruction which reads the AFCC register bit.

Make sure that pin 31, 32, 33 or 34 are configured as AFC function before you can use them (see chapter 9).

VLVL

VSYNC input pin (#57) level.

(case 1)

If VSYNC polarity is programmed to active high (see section 17.4 for detail), then when

VLVL = 0 means VSYNC is at low level, i.e., raster scan period

VLVL = 1 means VSYNC is at high level, i.e., vertical fly back period

(case 2)

If VSYNC polarity is programmed to active low,

VLVL = 0 means VSYNC is at low level, i.e., vertical fly back period.

VLVL = 1 means VSYNC is at high level, i.e., raster scan period.

## 13. VSYNC INTERRUPT & LEVEL STATUS BIT

VSYNC can generate an interrupt to the CPU if enabled. Vector address 0063H.

VINT SFR is the register to indicate the level of VSYNC pin as well as generating the interrupt to the CPU.

VINT (C9H)

bit7	bit6	bit5	bit4
—	—	—	—
bit3	bit2	bit1	bit0
—	—	—	VLVL

### 13.1 Interrupt Request – IQ7

IQ7 is the bit 5 of the SFR IRQ1 (address C0H).

IQ7

a leading edge of the active VSYNC generates a pending interrupt to the CPU and makes IQ7 bit high. In the service routine, this bit should be cleared before return to main routine. As long as this bit is high, a pending interrupt is always there. Each time VSYNC is active by the leading edge IQ7 bit is set to high. If the interrupt is not served by next leading VSYNC edge, the IQ7 is written to high again and no error of overrun is indicated.

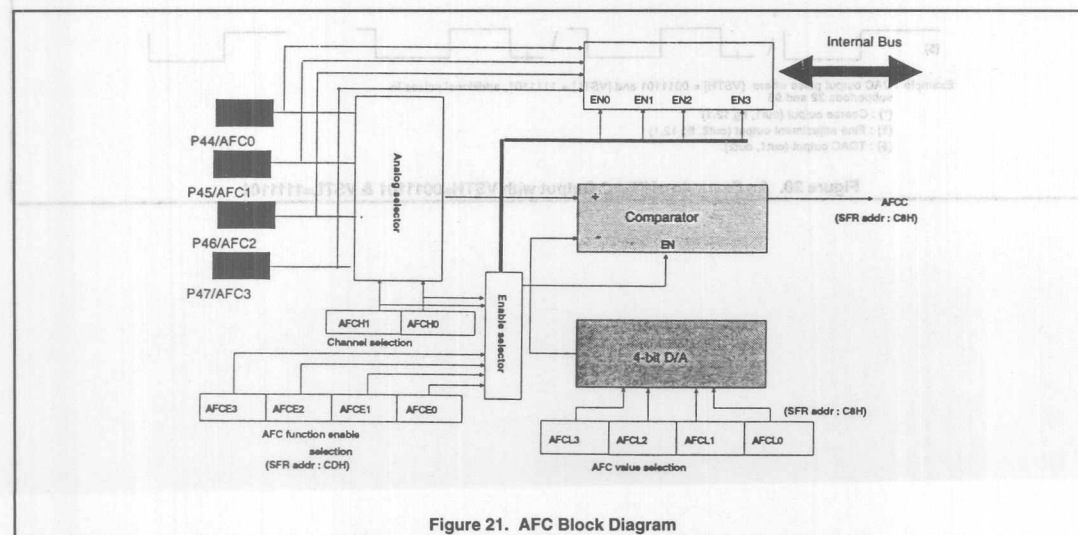


Figure 21. AFC Block Diagram

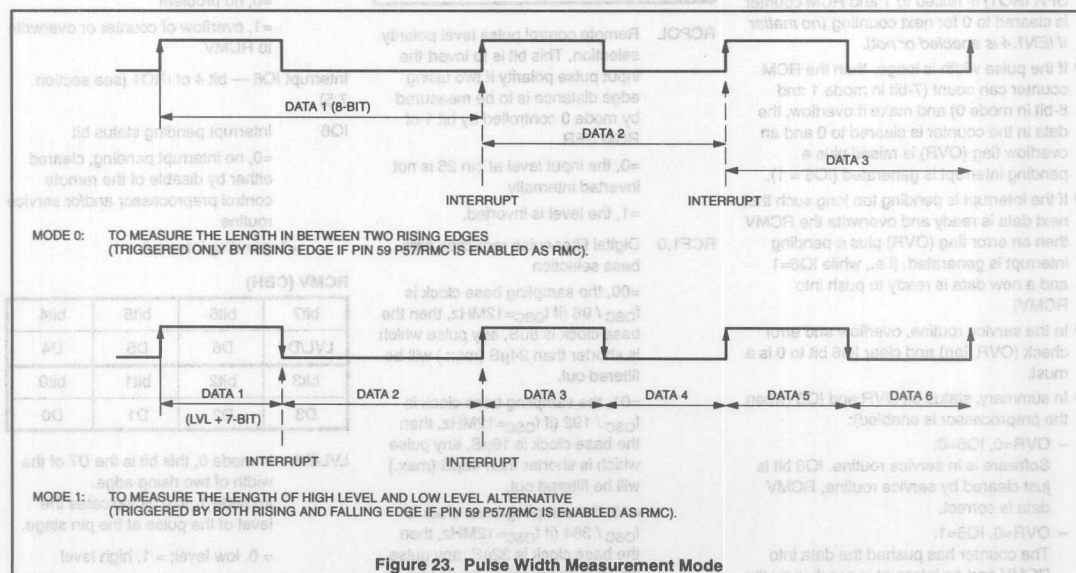
P83CL168/167/268/267

### 14.1 Block Diagram

Figure 22 shows the block diagram

**Figure 22. Remote Control Pre-Processor Block Diagram**

Figure 22. Remote Control Pre-Processor Block Diagram



**Figure 23. Pulse Width Measurement Mode**

## 14.2 The Operation

- To start the remote control preprocessor, SFR address CEH, the P5SEL register's bit 7 must be programmed to 1.
- The remote control pulse can be inverted or gone through for next stage processing at the input pin stage (polarity selection, bit <MU>6<D> of RCM SFR, see section 15.3).
- The digital filter and sampling counter measures the pulse width of the following two conditions
  - Mode 0:  
In between two rising edge, the pulse is sampled after the polarity selection circuitry shown in Figure 22 – RCM counter is 8-bit long.
  - Mode 1:  
In between rising and falling edge (high level) and then falling edge to next rising edge (low level) — RCM counter is 7-bit long. The MSB (8th bit) indicates the high (=1) or low (=0) level of the pulse at the input pin stage (non inverted, i.e., the level at the RMC pin, not after the polarity selection).
- After a complete pulse width is measured, the data stored in the RCM counter is pushed into a SFR called RCMV (address CBH) and a pending interrupt is generated to the CPU. Status flag IQ6 (the bit 4 of SFR IRQ1) is raised to 1 and RCM counter is cleared to 0 for next counting (no matter if IEN1.4 is enabled or not).
- If the pulse width is longer than the RCM counter can count (7-bit in mode 1 and 8-bit in mode 0) and make it overflow, the data in the counter is cleared to 0 and an overflow flag (OVR) is raised plus a pending interrupt is generated (IQ6 = 1).
- If the interrupt is pending too long such that next data is ready and overwrite the RCMV then an error flag (OVR) plus a pending interrupt is generated. (i.e., while IQ6=1 and a new data is ready to push into RCMV)
- In the service routine, overflow and error check (OVR flag) and clear IQ6 bit to 0 is a must.
- In summary, status bit OVR and IQ6 (when the preprocessor is enabled):
  - OVR=0, IQ6=0:  
Software is in service routine. IQ6 bit is just cleared by service routine, RCMV data is correct.
  - OVR=0, IQ6=1:  
The counter has pushed the data into RCMV and an interrupt is pending by the CPU.

- OVR=1, IQ6=1:  
A pending interrupt to the CPU indicates error happened either by overflow in the counter and/or overwrite of previous data in the RCMV. OVR should be cleared by software to ensure further operation.
- OVR=1, IQ6=0:  
Software in service routine. IQ6 bit is cleared by service routine. RCMV data is not correct. OVR should be cleared by software to ensure further operation.
- If the P5SEL.7 bit is programmed to 0,
  - the remote control preprocessor is stopped,
  - the data in the counter is cleared to 0,
  - the pending interrupt status bit (IQ6) and overflow/overwrite flag (OVR) are cleared
  - and pin 26 acts as P57 I/O port line.

## 14.3 Control, Status and Data Registers

### RCM (CAH)

bit7	bit6	bit5	bit4
—	RCPOL	RCF1	RCF0
bit3	bit2	bit1	bit0
RCS1	RCS0	RCM	OVR

**RCPOL** Remote control pulse level polarity selection. This bit is to invert the input pulse polarity if two falling edge distance is to be measured by mode 0 controlled by bit 1 of RCM SFR.

=0, the input level at pin 26 is not inverted internally  
=1, the level is inverted.

**RCF1,0** Digital filter noise reduction time base selection

=00, the sampling base clock is  $f_{osc} / 96$  (if  $f_{osc}=12\text{MHz}$ , then the base clock is 8 $\mu\text{S}$ , any pulse which is shorter than 24 $\mu\text{S}$  (max.) will be filtered out.

=01, the sampling base clock is  $f_{osc} / 192$  (if  $f_{osc}=12\text{MHz}$ , then the base clock is 16 $\mu\text{S}$ , any pulse which is shorter than 48 $\mu\text{S}$  (max.) will be filtered out.

=10, the sampling base clock is  $f_{osc} / 384$  (if  $f_{osc}=12\text{MHz}$ , then the base clock is 32 $\mu\text{S}$ , any pulse which is shorter than 96 $\mu\text{S}$  (max.) will be filtered out.

=11, the sampling base clock is  $f_{osc} / 768$  (if  $f_{osc}=12\text{MHz}$ , then the base clock is 64 $\mu\text{S}$ , any pulse which is shorter than 192 $\mu\text{S}$  (max.) will be filtered out.

**RCS1,0** Remote control pulse width measurement time base selection

=00, the sampling base clock is  $f_{osc} / 192$  (if  $f_{osc}=12\text{MHz}$ , then the base clock is 16 $\mu\text{S}$ )

=01, the sampling base clock is  $f_{osc} / 384$  (if  $f_{osc}=12\text{MHz}$ , then the base clock is 32 $\mu\text{S}$ )

=10, the sampling base clock is  $f_{osc} / 768$  (if  $f_{osc}=12\text{MHz}$ , then the base clock is 64 $\mu\text{S}$ )

=11, the sampling base clock is  $f_{osc} / 1536$  (if  $f_{osc}=12\text{MHz}$ , then the base clock is 128 $\mu\text{S}$ )

**RCM** Measurement mode selection

=0 (mode 0), measuring of the pulse width of two rising edge at the point after polarity selection circuitry. Counter is 8-bit long.

=1 (mode 1), measurement of the pulse width from rising edge to falling edge, counter is 7-bit long

**OVR** Overflow/overwrite flag

=0, no problem

=1, overflow of counter or overwrite to RCMV

**Interrupt IQ6** — bit 4 of IRQ1 (see section 7.5)

**IQ6** Interrupt pending status bit

=0, no interrupt pending, cleared either by disable of the remote control preprocessor and/or service routine

=1, pending interrupt

### RCMV (CBH)

bit7	bit6	bit5	bit4
LVL/D7	D6	D5	D4
bit3	bit2	bit1	bit0
D3	D2	D1	D0

**LVL/D7** in mode 0, this bit is the D7 of the width of two rising edge.

in mode 1, this bit indicates the level of the pulse at the pin stage.  
= 0, low level; = 1, high level

**D6 .. 0** the lower 7 bits of the pulse width either in mode 0 or 1.

# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

## P83CL168/167/268/267

### 15. HIGH SPEED 8-BIT, 3-WIRE SERIAL I/O

The high speed 8-bit, 3-wire serial I/O features 1Mega baud rate maximum speed for very high speed data transmission in between CPU and other peripheral devices. Interrupt is generated whenever a completed byte has been sent or received. In the data transmission process, LSB first or MSB first can be selected by software.

#### 15.1 The Block Diagram

Figure 24 shows the block diagram. Clock can be selected from 1M, 500K, 250K (@fosc=12MHz) and external source.

#### 15.2 The Control Register

##### HIOD (C6H)

bit7	bit6	bit5	bit4
D7	D6	D5	D4
bit3	bit2	bit1	bit0
D3	D2	D1	D0

HIOD is the 8-bit data register of the high speed 8-bit serial I/O.

##### HIOCON (C7H)

bit7	bit6	bit5	bit4
—	—	—	CLKE
bit3	bit2	bit1	bit0
DOUTE	CLK1	CLK0	DIR

CLKE clock pin Hi-Z or enable selection (if pin 28 is selected as CLK function)  
=0, Hi-Z  
=1, CLK pin is driven by the high speed I/O's clock signal

DOUTE data out pin Hi-Z selection (if pin 20 is selected as Dout function)  
=0, Hi-Z  
=1, Dout pin is driven by the high speed I/O's data signal

CLK1 .. 0 clock source selection  
=00, fosc/12  
(=1MHz @fosc=12MHz)  
=01, fosc/24  
=10, fosc/48  
=11, external source from CLK pin

##### DIR

Shift direction control

When in the shift out operation  
=0, MSB first  
=1, LSB first

When in the shift in operation  
=1, first bit becomes the MSB of the data byte  
=0, first bit becomes the LSB of the data byte

Interrupt IQ5 - bit 3 of IRQ1 SFR

IQ5 High speed I/O interrupt status bit, raises to '1' when a complete byte has been sent or received.

=1, a pending interrupt  
=0, no pending interrupt

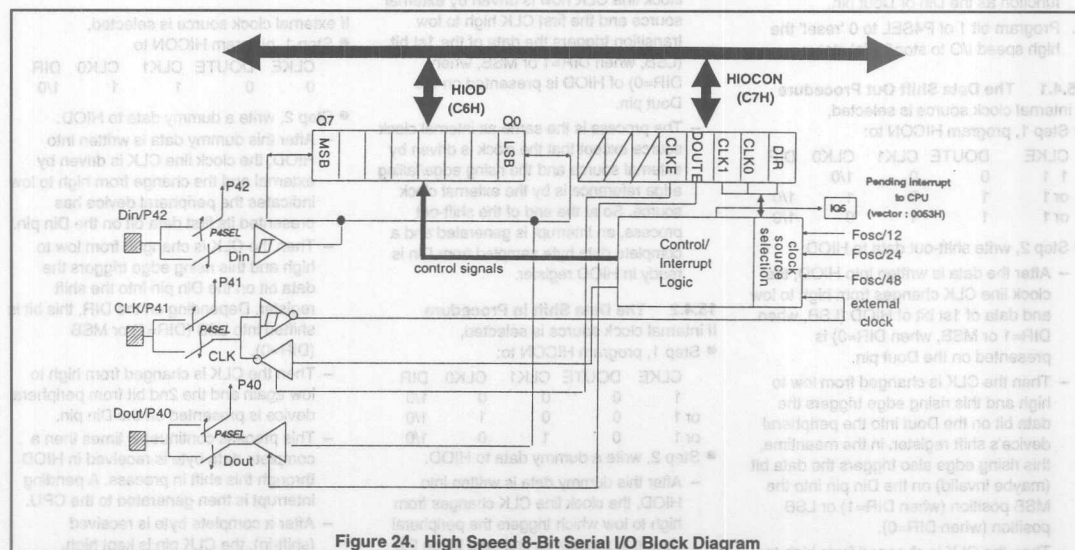


Figure 24. High Speed 8-Bit Serial I/O Block Diagram



# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

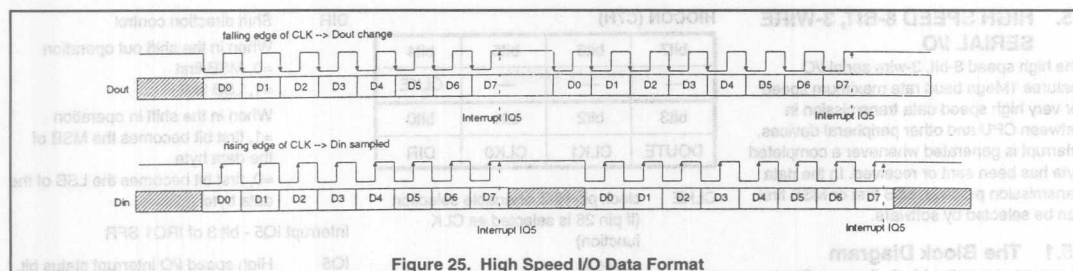


Figure 25. High Speed I/O Data Format

## 15.3 The Shift-In/Shift-Out Format

Figure 25 shows the shift-in and shift-out data format. The falling edge of the clock (CLK) is the transition point of the Dout. The rising edge of the clock (CLK) samples the high/low level of the data on the Din line.

## 15.4 How to Program the High Speed I/O

- The operation to program bit 1 of SFR P4SEL (addr: CDH) to 1 is to enable the high speed I/O to standby status (ready for the shift-in or shift-out). In the mean time, one (or both) of the bit2 and bit 0 of P4SEL can be programmed to 1 to function as the Din or Dout pin.
- Program bit 1 of P4SEL to 0 'reset' the high speed I/O to stop/initial status.

### 15.4.1 The Data Shift Out Procedure

If internal clock source is selected,

- Step 1, program HICON to:

CLKE	DOUT	CLK1	CLK0	DIR
1	0	0	1/0	
or 1	1	0	1	1/0
or 1	1	1	0	1/0

- Step 2, write shift-out data to HIOD.

- After the data is written into HIOD, the clock line CLK changes from high to low and data of 1st bit of HIOD (LSB, when DIR=1 or MSB, when DIR=0) is presented on the Dout pin.
- Then the CLK is changed from low to high and this rising edge triggers the data bit on the Dout into the peripheral device's shift register. In the meantime, this rising edge also triggers the data bit (maybe invalid) on the Din pin into the MSB position (when DIR=1) or LSB position (when DIR=0).
- Then the CLK is changed from high to low again and the 2nd bit is presented on the Dout pin.
- This process continues 8 times when a complete data byte is sent through this shift out process. A pending interrupt is then generated to the CPU and a

complete byte is ready and stored in the SFR HIOD which is the data sampled from Din. This data may be of no use and is discarded by the software.

- After a complete byte is sent (shift-out), the CLK pin is kept high and Dout is having the last data bit presented.
- Write to HIOD again is the way to restart next byte transmission.

If external clock source is selected,

- Step 1, program HICON to

CLKE	DOUT	CLK1	CLK0	DIR
0	1	1	1	1/0

- Step 2, write shift-out data to HIOD.

- After the data is written into HIOD, the clock line CLK now is driven by external source and the first CLK high to low transition triggers the data of the 1st bit (LSB, when DIR=1 or MSB, when DIR=0) of HIOD is presented on the Dout pin.
- The process is the same as internal clock source except that the clock is driven by external source and the rising edge/falling edge reference is by the external clock source. So at the end of the shift-out process, an interrupt is generated and a complete data byte sampled from Din is ready in HIOD register.

### 15.4.2 The Data Shift In Procedure

If internal clock source is selected,

- Step 1, program HICON to:

CLKE	DOUT	CLK1	CLK0	DIR
1	0	0	0	1/0
or 1	0	0	1	1/0
or 1	0	1	0	1/0

- Step 2, write a dummy data to HIOD.

- After this dummy data is written into HIOD, the clock line CLK changes from high to low which triggers the peripheral device to present its first data bit on the Din pin.
- Then the CLK is changed from low to high and this rising edge triggers the data bit on the Din pin into the shift register. Depending on the DIR, this bit is shifted into LSB (DIR=1) and it will become the MSB of the received byte; or

MSB (DIR=0) and it will become the LSB of the received byte.

- Then the CLK is changed from high to low again and the 2nd bit from peripheral device is presented on the Din pin.
- This process continues 8 times then a complete data byte is received in HIOD through this shift in process. A pending interrupt is then generated to the CPU.
- After a complete byte is received (shift-in), the CLK pin is kept high.
- Dout pin in the whole process is in tri-state status if that pin is enabled as the Dout pin function.
- Write to HIOD again is the way to restart next byte transmission.

If external clock source is selected,

- Step 1, program HICON to

CLKE	DOUT	CLK1	CLK0	DIR
0	0	1	1	1/0

- Step 2, write a dummy data to HIOD.

- After this dummy data is written into HIOD, the clock line CLK is driven by external and the change from high to low indicates the peripheral device has presented its first data bit on the Din pin.
- Then the CLK is changed from low to high and this rising edge triggers the data bit on the Din pin into the shift register. Depending on the DIR, this bit is shifted into LSB (DIR=1) or MSB (DIR=0).
- Then the CLK is changed from high to low again and the 2nd bit from peripheral device is presented on the Din pin.
- This process continues 8 times then a complete data byte is received in HIOD through this shift in process. A pending interrupt is then generated to the CPU.
- After a complete byte is received (shift-in), the CLK pin is kept high.
- Dout pin in the whole process is in tri-state status if that pin is enabled as the Dout pin function.
- Write to HIOD again is the way to restart next byte transmission. Before this operation is done, any changes on the CLK/Din pins are ignored.

# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

## 16. OSD (ON SCREEN DISPLAY) FUNCTION

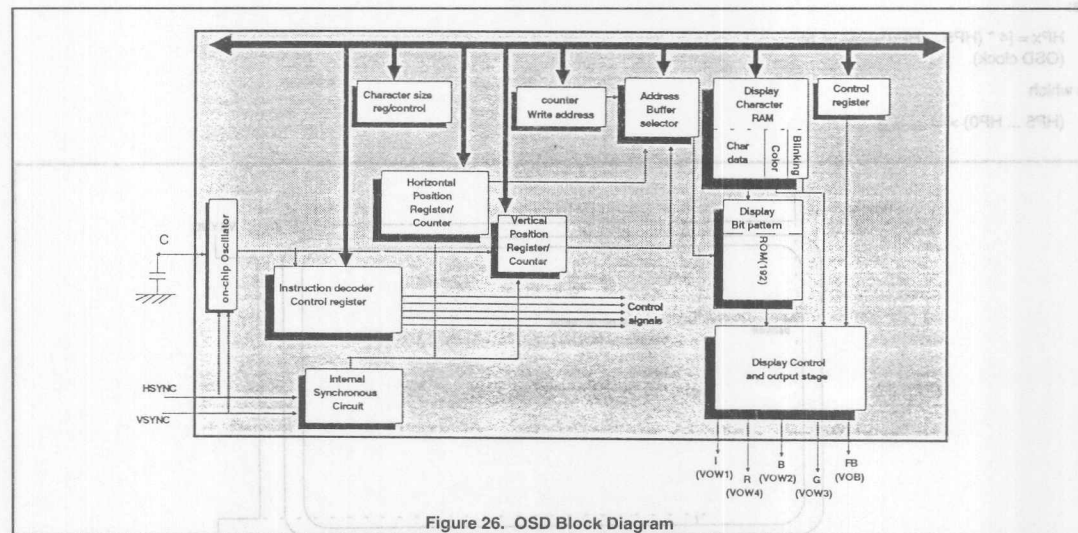
### 16.1 Features

- Programmable VSYNC/HSYNC active level
- Display RAM: 160 × 12 bit
- Display character fonts: 128 (126 customer font + 2 reserved)
- One vertical display starting position counter 64 different starting positions controlled by software.
- One horizontal display starting position control for OSD. 64 different positions controlled by software.
- Character size: 4 different character size, line-by-line basis, 1 horizontal dot = 1 or 2 OSD clock period. 1 vertical dot = 1 or 2 horizontal scan lines. Individually controllable.

- Character matrix: 12 × 18 with no spacing between characters
- Foreground colors: 16, character-by-character basis
- Background/shadowing modes:
  - Two primary modes: TV mode or Monitor mode, frame basis. In each group, three submodes are available.
  - Sub-mode 1: superimpose (no background),
  - Sub-mode 2: north-west shadowing,
  - Sub-mode 3: box shadowing,
  - all these three are character row basis.
- Background colors: 16, word-by-word basis. Available when background is either in "north-west shadowing", "box

shadowing" and "frame shadowing" submodes.

- Display RAM starting address is programmable. Fast switching between bank of display characters are possible through software control.
- PLL oscillator for OSD function with software selectable OSD clock period (e.g., scalable character width by software)
- Character blinking: 1:1 (frequency : Fv/32, Fv/64, programmable), character basis
- Display format: Flexible display format by using CR (carriage return) and SP1, SP2 (space) code.
- Display RAM address post increment each time a new data is written into.
- Vertical jitter cancelling circuit to avoid unstable VSYNC trailing edge mismatch HSYNC signal.
- OSD clock operating range: 4 – 10MHz





# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

## 16.2 Horizontal and Vertical Display Position Control

### OSORGH (C3H)

bit7	bit6	bit5	bit4
—	—	HP5	HP4
bit3	bit2	bit1	bit0
HP3	HP2	HP1	HP0

The horizontal position counter is increased every OSDC (OSD clock) cycle after the programmed level of HSYNCH occurs at the HSYNC terminal (pin) and is reset when the opposite polarity of the HSYNC is reached. Horizontal start position is controlled by SFR OSORGH which is a 6-bit register (Figure 27).

Reference point: trailing edge of HSYNC.

The horizontal starting position is calculated as:

$$HPx = [4 * (HP5 \dots HP0)_{\text{decimal}} + 5] \text{ (OSD clock)}$$

in which

$$(HP5 \dots HP0) \geq 4 \text{ decimal}$$

### OSORGV (C2H)

bit7	bit6	bit5	bit4
—	—	VP5	VP4
bit3	bit2	bit1	bit0
VP3	VP2	VP1	VP0

The vertical position counter is increased every HSYNC cycle and is reset by the VSYNC signal.

Vertical start position is controlled by SFR OSORGV which is a 6-bit register too. Reference point: leading edge of VSYNC.

The starting position is calculated as

$$VPx = [4 * (VP5 \dots VP0)_{\text{decimal}}] \text{ (horizontal scan lines)}$$

in which

$$(VP5 \dots VP0)_{\text{decimal}} > 0$$

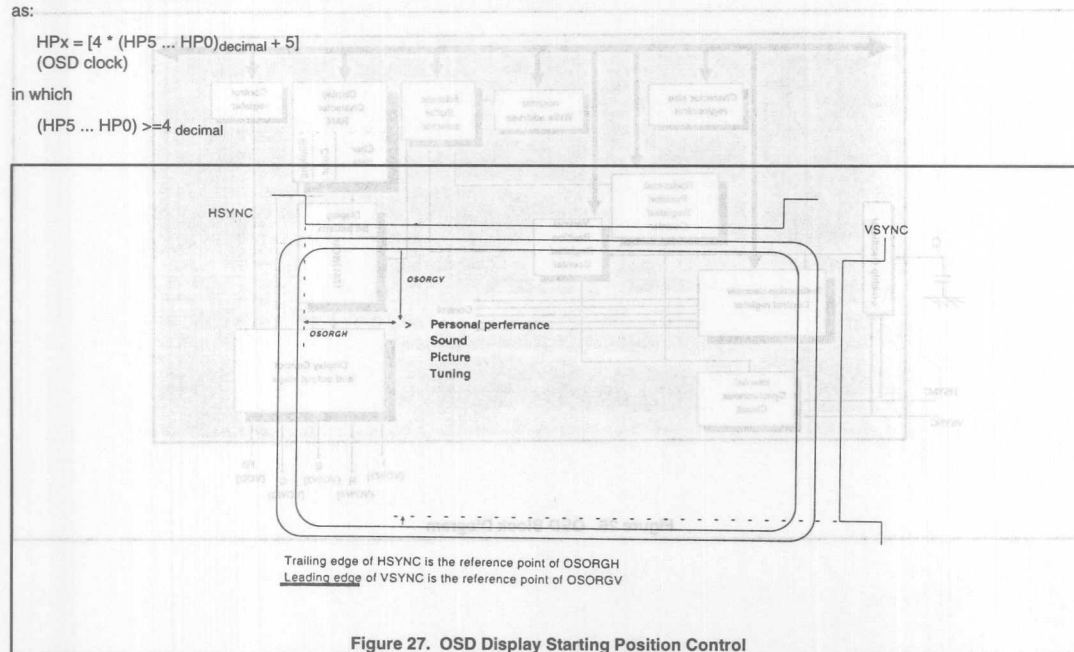


Figure 27. OSD Display Starting Position Control

Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

### 16.3 OSD Clock Generator

The clock generator is made of a PLL (Phase Lock Loop) circuitry. The VCO (Voltage Controlled Oscillator) outputs a constant clock with frequency ranges in between 4MHz to 10MHz.

The input frequency, F1, is the horizontal sync (HSYNC) frequency. Depending on the value programmed in the 6-bit counter, the

$$F_{vco} = F_1 * 16 * [\text{value of 6 bit counter}]$$

with

$$16_{\text{decimal}} < [\text{value of 6 bit counter}] < 40_{\text{decimal}}$$

The value 16 is the 4 bit prescaler which gives the output of the VCO to increase/decrease in step of  $16 \cdot F_1$  Hz. Given an example of horizontal frequency 15.750KHz, The  $F_{vco}$  is then increased or

decreased in steps of  $16 \cdot 15.750\text{KHz} = 252\text{KHz} = 0.25\text{MHz}$  step.

The Fvco is then feed into the buffer to generate the OSDC (OSD dot clock frequency) signal which gives the OSDC in range of 4 – 10MHz. Decreasing the OSDC frequency gives broader characters. Typical recommended OSD clock frequency is 6 – 8MHz.

The VCO is synchronized with the high-to-low edge of the F2 clock (always the trailing edge of HSYNC signal). The programmable active level detector is to either pass the HSYNC signal (if HSYNC is active high) or to invert the HSYNC signal (if the HSYNC is active low). It makes F2 always active high.

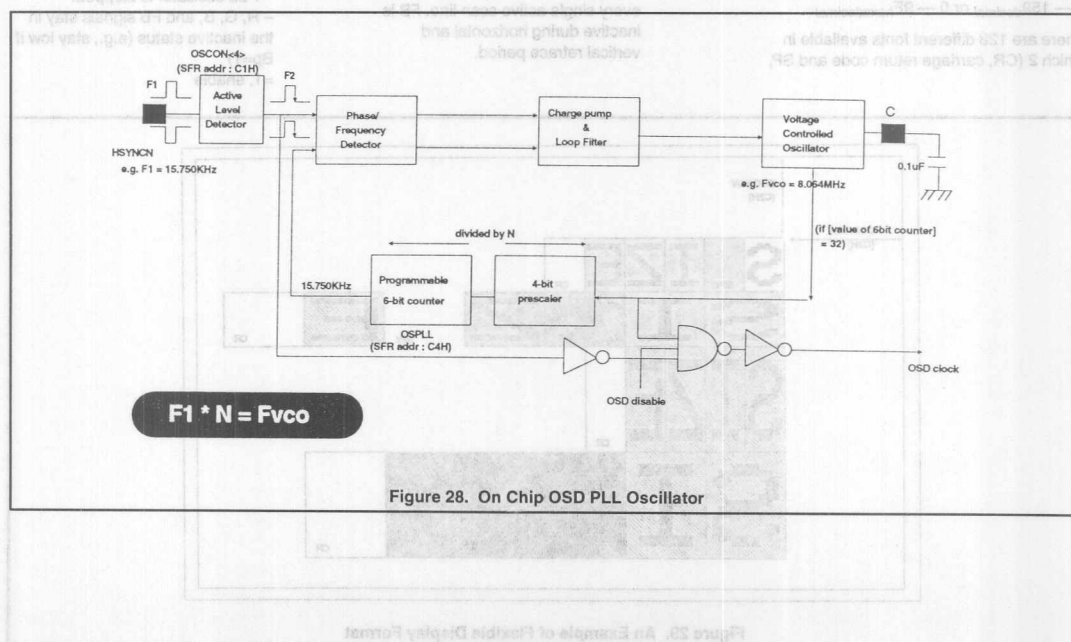
Figure 28 illustrates the block diagram of on chip OSD clock generator circuitry.

The on chip OSD PLL oscillator will not stop, as long as it is enabled by bit 0, the OSDE bit, of SFR OSCON (C1H). The following description in V1.1 should then be discarded totally.

Because no large current is sourced or sunk in and out of the clock pin as the RC or LC type of oscillator normally is, the radiation generated by the OSD clock source can be reduced dramatically.

The 6 bit PLL programmable counter data register is in C4H.

OSPLL (C4H)			
bit7	bit6	bit5	bit4
—	—	PLL5	PLL4
bit3	bit2	bit1	bit0
PLL3	PLL2	PLL1	PLL0



**Figure 28. On Chip OSD PLL Oscillator**

## Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

### 16.4 Flexible On Screen Display Format

Figure 29 shows a typical example of OSD generated by the P83CL168. The vertical and horizontal starting position is controlled by two separate SFR OSORGH and OSORGV. There are 160 display RAM in total where they are considered as a linear addressed RAM. The first character is from the display RAM address pointed by OSSTART (SFR address C5H).

#### OSSTART (C5H)

bit7	bit6	bit5	bit4
START7	START6	START5	START4
bit3	bit2	bit1	bit0
START3	START2	START1	START0

The display RAM address is from 0 — 159 decimal or 0 — 9F hexadecimal.

There are 128 different fonts available in which 2 (CR, carriage return code and SP,

space code) of them are reserved for special function. Each time a CR code is met, like in the computer application, the display changes to next row. A space code is meant to change the background color of the characters followed (see section 16.5 for further detail).

OSCON is the OSD control register.

#### OSCON (C1H)

bit7	bit6	bit5	bit4
—	—	Mode	Hp
bit3	bit2	bit1	bit0
Vp	Bp	BF	OSDE

**Mode** Background mode selection  
 =0, TV mode (FB only active when there are characters displayed)  
 =1, Monitor mode (FB is active in every single active scan line. FB is inactive during horizontal and vertical retrace period.

**Hp** HSYNC active polarity selection

=0, active low HSYNC input

=1, active high HSYNC input

(Figure 30)

**Vp**

VSYNC active polarity selection

=0, active low VSYNC input

=1, active high VSYNC input

(Figure 30)

**Bp**

Active R, G, B, and FB output

polarity selection

=0, active low

=1, active high (Figure 31)

**BF**

Character blinking frequency

control (duty cycle is fixed to 1 : 1)

=0, <N>Frequency is  $F_v/32$  where

$F_v$  is the VSYNC frequency

=1, Frequency is  $F_v/64$ .

**OSDE**

OSD circuit general enable/disable

=0, disable

— PLL oscillator is stopped.

— R, G, B, and FB signals stay in the inactive status (e.g., stay low if Bp=1)

=1, enable

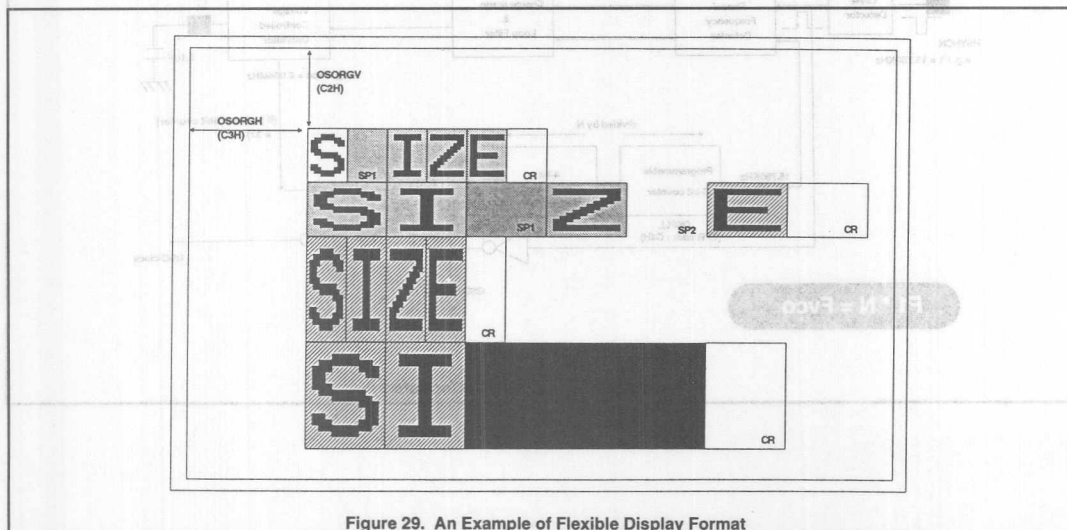


Figure 29. An Example of Flexible Display Format

# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

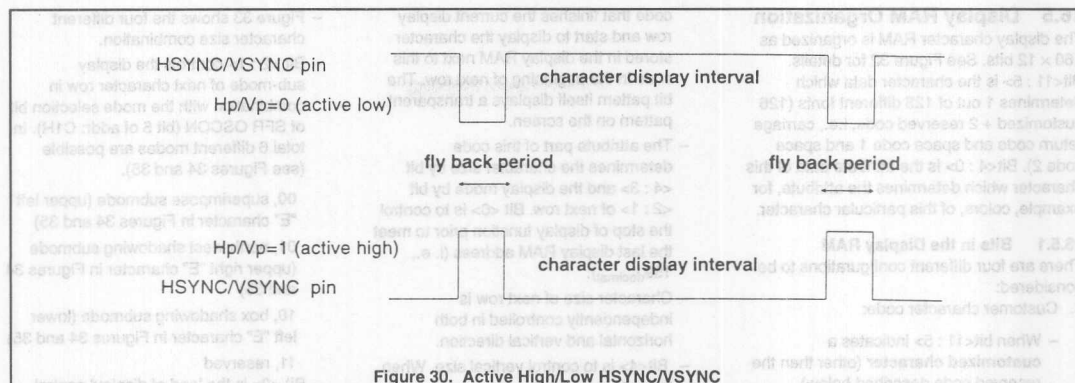


Figure 30. Active High/Low HSYNC/VSYNC

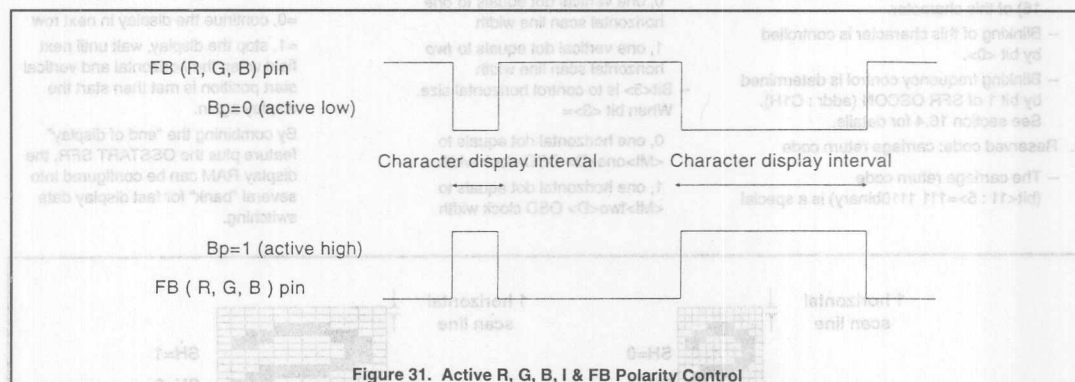


Figure 31. Active R, G, B, I &amp; FB Polarity Control

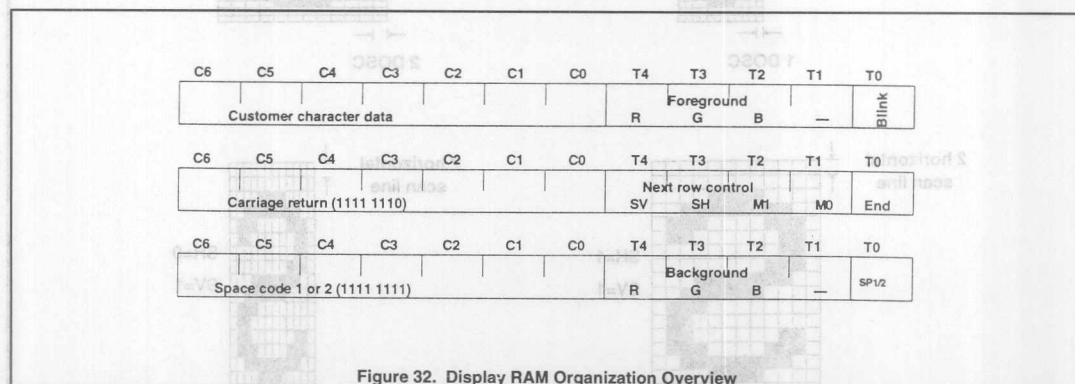


Figure 32. Display RAM Organization Overview

# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

## 16.5 Display RAM Organization

The display character RAM is organized as 160 × 12 bits. See Figure 32 for details.

Bit<11 : 5> is the character data which determines 1 out of 128 different fonts (126 customized + 2 reserved code, i.e., carriage return code and space code 1 and space code 2). Bit<4 : 0> is the attribute data of this character which determines the attribute, for example, colors, of this particular character.

### 16.5.1 Bits in the Display RAM

There are four different configurations to be considered:

#### 1. Customer character code:

- When bit<11 : 5> indicates a **customized character** (other than the reserved code described below), bit<4 : 1> determines the color (1 out of 16) of this character.
- Blinking of this character is controlled by bit<0>.
- Blinking frequency control is determined by bit 1 of SFR OSCON (addr : C1H). See section 16.4 for details.

#### 2. Reserved code: carriage return code

- The carriage return code (bit<11 : 5>=111 1110binary) is a special

code that finishes the current display row and start to display the character stored in the display RAM next to this code in the beginning of next row. The bit pattern itself displays a transparent pattern on the screen.

- The attribute part of this code determines the character size by bit<4 : 3> and the display mode by bit<2 : 1> of next row. Bit<0> is to control the stop of display function prior to meet the last display RAM address (i.e., 159<sub>decimal</sub>).
- Character size of next row is independently controlled in both horizontal and vertical direction.
- Bit<4> is to control vertical size. When bit<<4> =

0, one vertical dot equals to one horizontal scan line width

1, one vertical dot equals to two horizontal scan line width

- Bit<3> is to control horizontal size. When bit<3>=

0, one horizontal dot equals to <Ml>one<D> OSD clock width

1, one horizontal dot equals to <Ml>two<D> OSD clock width

- Figure 33 shows the four different character size combination.

- Bit<2 : 1> controls the display sub-mode of next character row in combination with the mode selection bit of SFR OSCON (bit 5 of addr: C1H). In total 6 different modes are possible (see Figures 34 and 35).

00, superimpose submode (upper left "E" character in Figures 34 and 35)

01, north-west shadowing submode (upper right "E" character in Figures 34 and 35)

10, box shadowing submode (lower left "E" character in Figures 34 and 35)

11, reserved

- Bit<0> is the 'end of display' control before running out of display RAM

=0, continue the display in next row

=1, stop the display, wait until next field when the horizontal and vertical start position is met then start the display again.

By combining the "end of display" feature plus the OSSTART SFR, the display RAM can be configured into several "bank" for fast display data switching.

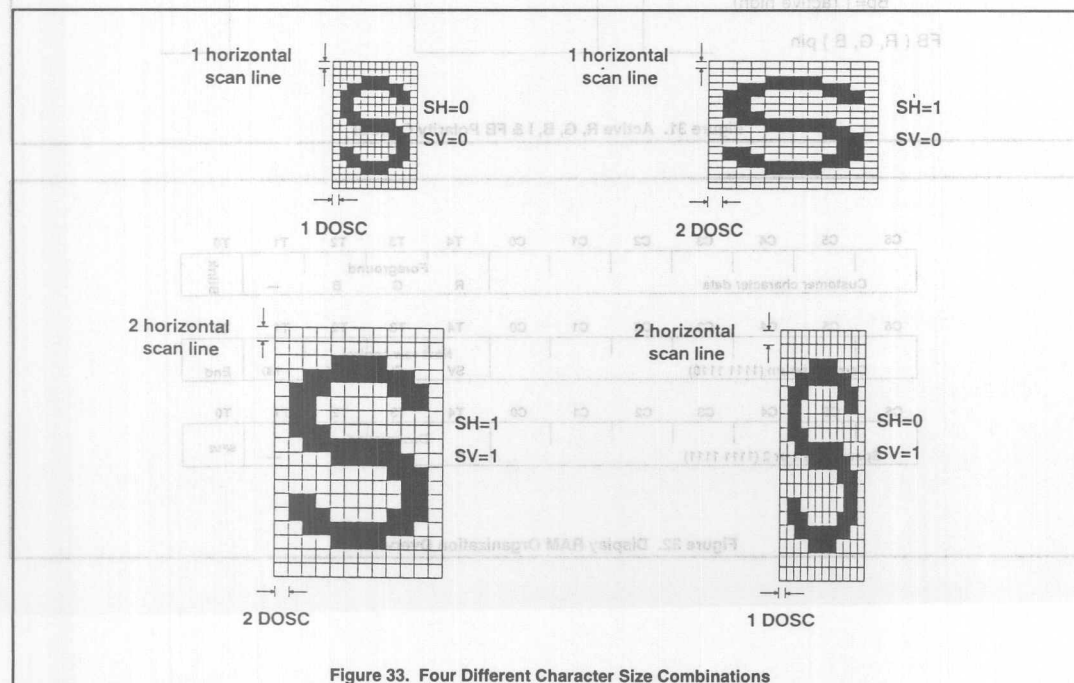


Figure 33. Four Different Character Size Combinations



# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

## 3. Reserved code:

Space code 1 and Space code 2

- Space code 1,  $C<7:0>=111\ 1111$ ,  $T<0>=0$ ,  $T<4:1>=$ background color control
- Space code 2,  $C<7:0>=111\ 1111$ ,  $T<0>=1$ ,  $T<4:1>=$ background color control

The character code  $C<7:0>=111\ 1101$  is for customer font usage.

- The background color is the color of the bit pattern in the following conditions, (see Figures 34 and 35)
- north-west shadowing submode, it is the color of the shadow bit pattern.

- box shadowing submode, it is the color of the dots in the  $12 \times 18$  character dot matrix box where no foreground character bits present.
- in the superimpose submode there is NO background color.

- Space code 1 & space code 2 are the codes which displays a transparent pattern on the screen and lasts for 1 character width in general.
- In the TV mode plus, (Figure 36)

superimpose submode:

SP1 and SP2 display exactly the same transparent pattern.

north-west shadowing submode:

SP1 and SP2 display exactly the same transparent pattern.

box shadowing submode:

SP2 displays a transparent pattern while SP1 displays a  $12 \times 18$  dots pattern filled with the background color.

- In the monitor mode, the background color of the monitor (not of the characters) is controlled by SFR OSCON2  $<3:0>$  (addr: CFH). The SP1 and SP2 behave the same as they are in the TV mode (Figure 37).

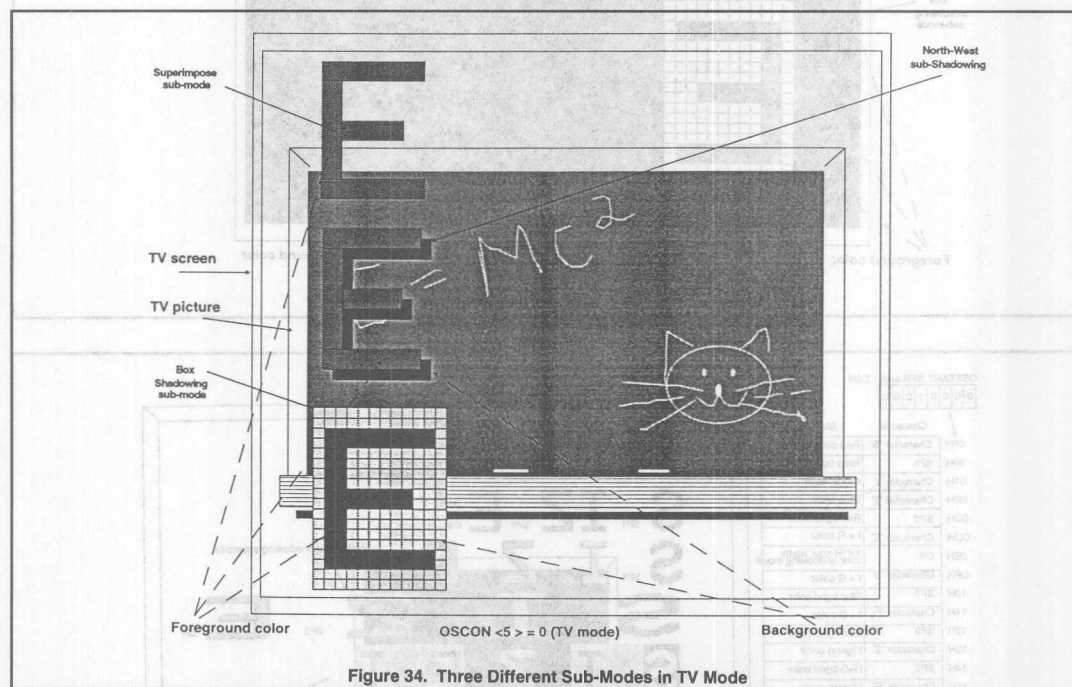


Figure 34. Three Different Sub-Modes in TV Mode



# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

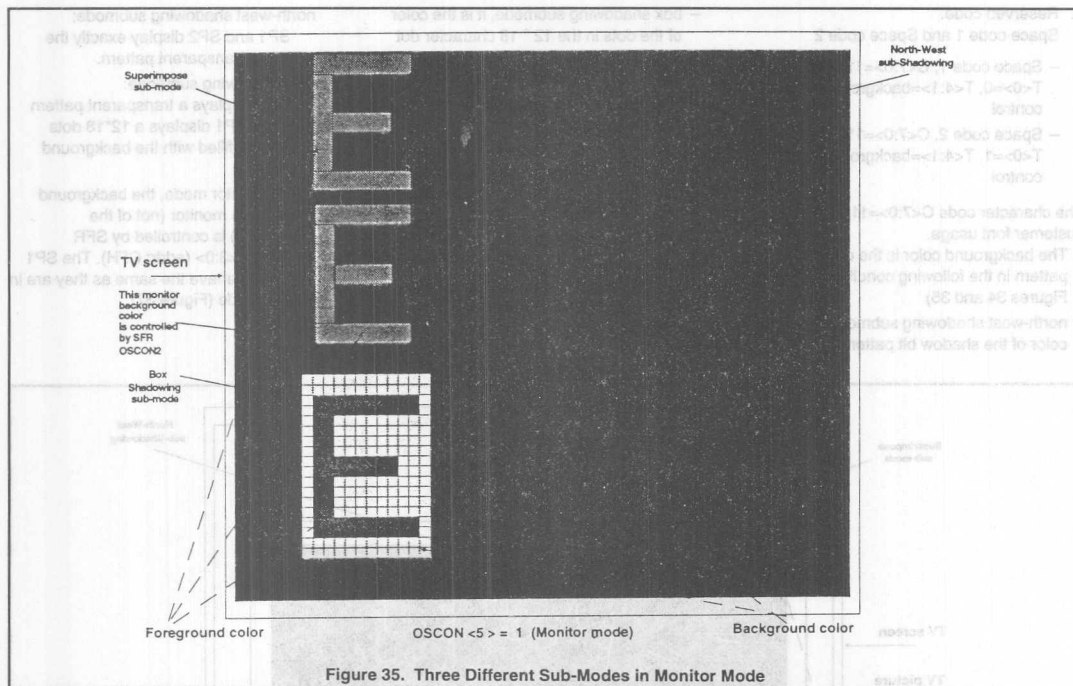


Figure 35. Three Different Sub-Modes in Monitor Mode

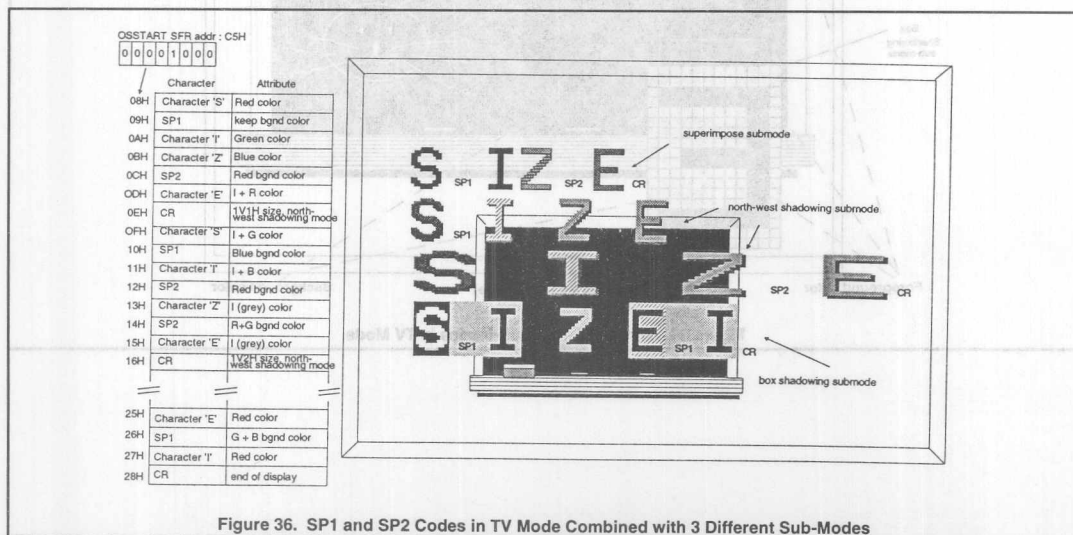


Figure 36. SP1 and SP2 Codes in TV Mode Combined with 3 Different Sub-Modes

# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

## P83CL168/167/268/267

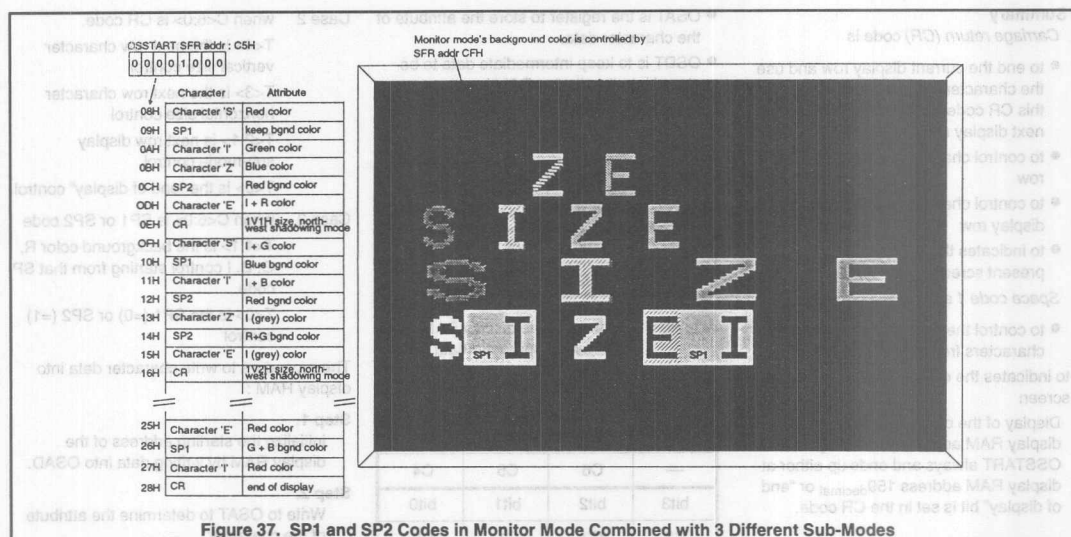
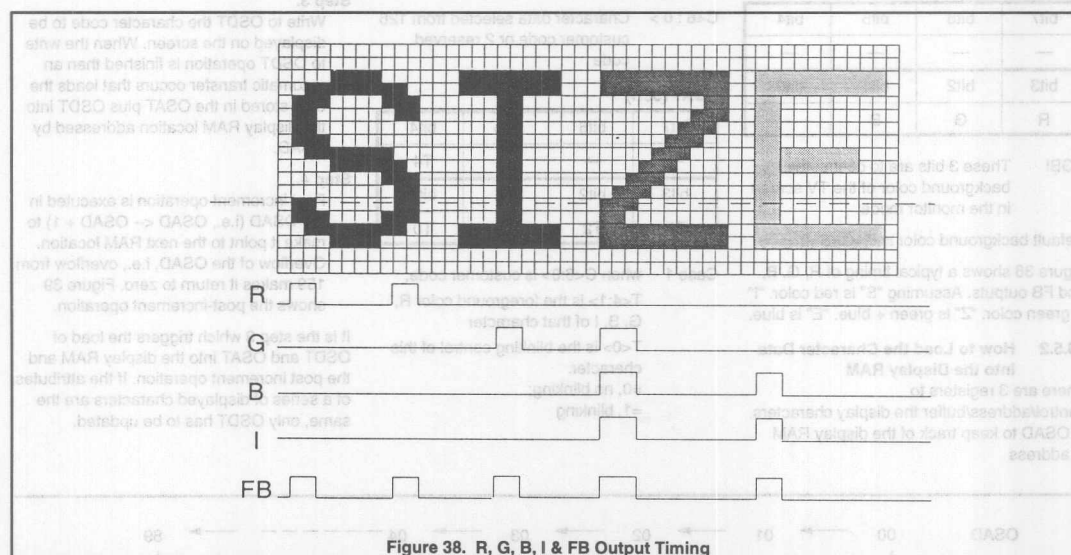


Figure 37. SP1 and SP2 Codes in Monitor Mode Combined with 3 Different Sub-Modes



# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

## P83CL168/167/268/267

### In Summary

#### 1. Carriage return (CR) code is

- to end the current display row and use the character in the display RAM next to this CR code as the first character of next display row.
- to control character size of next display row
- to control character submode of next display row
- to indicates the end of display for present screen

#### 2. Space code 1 and Space code 2 is

- to control the background color of the characters from this space code
- to indicates the end of display for present screen

- Display of the characters starts from display RAM address pointed by SFR OSSTART always and ends up either at display RAM address 159<sub>decimal</sub> or "end of display" bit is set in the CR code.

### OSCON2 (CFH)

bit7	bit6	bit5	bit4
—	—	—	—
bit3	bit2	bit1	bit0
R	G	B	—

RGBI These 3 bits are to control the background color of the TV screen in the monitor mode.

Default background color is R=G=B=0

Figure 38 shows a typical timing of R, G, B, and FB outputs. Assuming "S" is red color. "I" is green color. "Z" is green + blue. "E" is blue.

### 16.5.2 How to Load the Character Data into the Display RAM

- There are 3 registers to control/address/buffer the display characters.
- OSAD to keep track of the display RAM address

- OSAT is the register to store the attribute of the character data.

- OSDT is to keep intermediate data to be stored into the display RAM location addressed by the OSAD register.

### OSAD (9AH)

bit7	bit6	bit5	bit4
AD7	AD6	AD5	AD4
bit3	bit2	bit1	bit0
AD3	AD2	AD1	AD0

AD<7:0> Address of the display RAM where next character data is to be written into

### OSDT (99H)

bit7	bit6	bit5	bit4
—	C6	C5	C4
bit3	bit2	bit1	bit0
C3	C2	C1	C0

C<6:0> Character data selected from 126 customer code or 2 reserved code

### OSAT (98H)

bit7	bit6	bit5	bit4
—	—	—	T4
bit3	bit2	bit1	bit0
T3	T2	1	T0

- Case 1 when C<6:0> is customer code,  
T<4:1> is the foreground color R, G, B, I of that character  
T<0> is the blinking control of this character.  
=0, no blinking;  
=1, blinking

#### Case 2 when C<6:0> is CR code,

T<4> is the next row character vertical size control

T<3> is the next row character horizontal size control

T<2:1> is next row display sub-mode control

T<0> is the "end of display" control

#### Case 3 when C<6:0> is SP1 or SP2 code

T<4:1> is the background color R, G, B, I control starting from that SP code

T<0> is the SP1 (=0) or SP2 (=1) control

The procedure to write character data into display RAM :

#### Step 1.

initialize the starting address of the display RAM by writing data into OSAD.

#### Step 2.

Write to OSAT to determine the attribute of the character.

#### Step 3.

Write to OSDT the character code to be displayed on the screen. When the write to OSDT operation is finished then an automatic transfer occurs that loads the data stored in the OSAT plus OSDT into the display RAM location addressed by OSAD.

#### Step 4.

Post increment operation is executed in the OSAD (i.e., OSAD <-- OSAD + 1) to make it point to the next RAM location. Overflow of the OSAD, i.e., overflow from 159 makes it return to zero. Figure 39 shows the post-increment operation.

It is the step 3 which triggers the load of OSDT and OSAT into the display RAM and the post increment operation. If the attributes of a series of displayed characters are the same, only OSDT has to be updated.

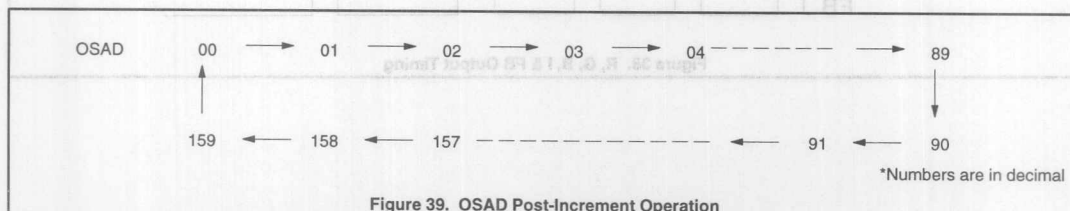
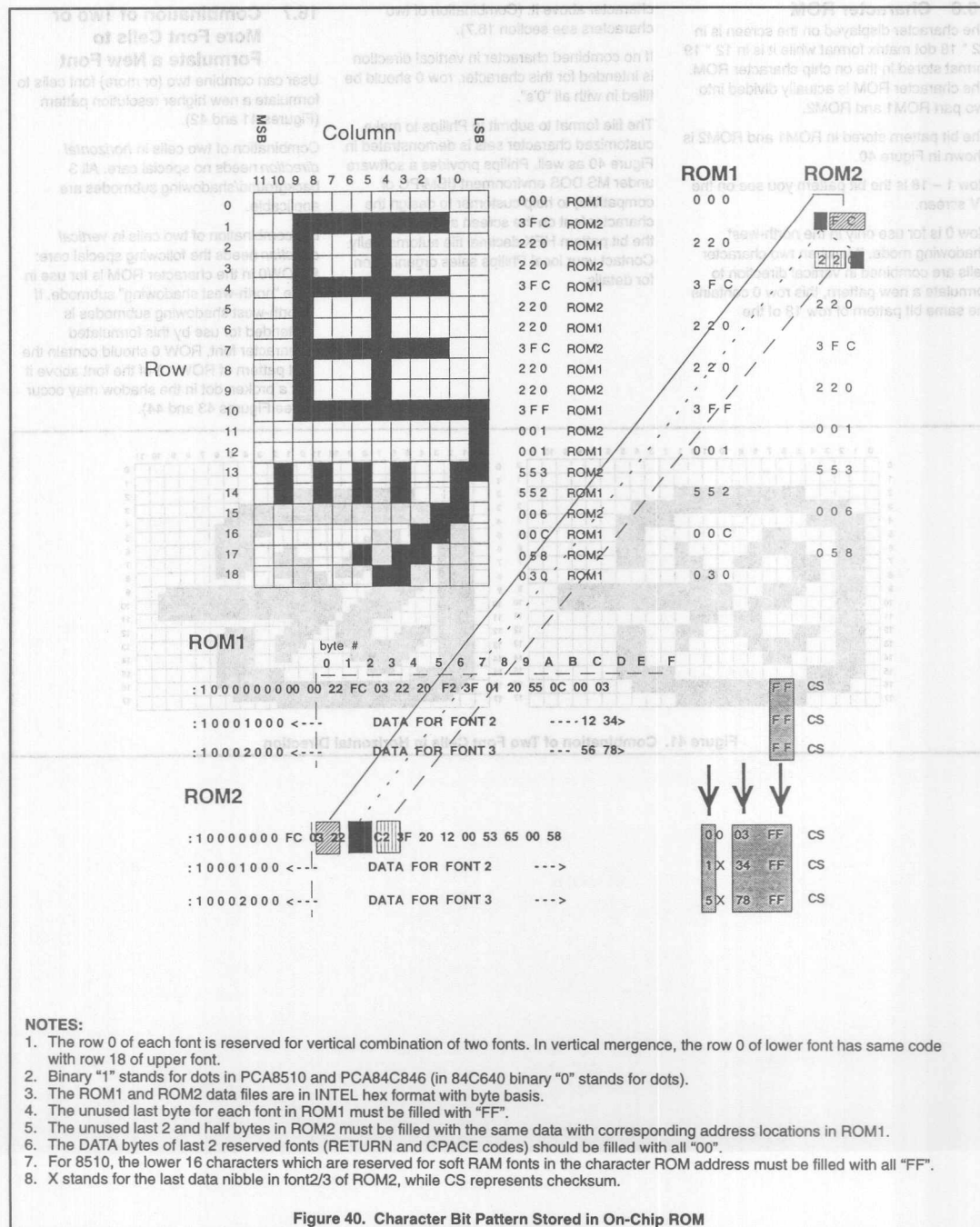


Figure 39. OSAD Post-Increment Operation

## Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267



## Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

### 16.6 Character ROM

The character displayed on the screen is in 12 \* 18 dot matrix format while it is in 12 \* 19 format stored in the on chip character ROM. The character ROM is actually divided into two part ROM1 and ROM2.

The bit pattern stored in ROM1 and ROM2 is shown in Figure 40.

Row 1 – 18 is the bit pattern you see on the TV screen.

Row 0 is for use only in the north-west shadowing mode. It is when two character cells are combined in vertical direction to formulate a new pattern, this row 0 contains the same bit pattern of row 18 of the

character above it. (Combination of two characters see section 16.7).

If no combined character in vertical direction is intended for this character, row 0 should be filled in with all "0's".

The file format to submit to Philips to make customized character sets is demonstrated in Figure 40 as well. Philips provides a software under MS DOS environment (IBM/PC or compatible) to help customer to design the character font on the screen and to generate the bit pattern HEX decimal file automatically. Contact your local Philips sales organization for details.

### 16.7 Combination of Two or More Font Cells to Formulate a New Font

User can combine two (or more) font cells to formulate a new higher resolution pattern (Figures 41 and 42).

Combination of two cells in *horizontal direction* needs no special care. All 3 background/shadowing submodes are applicable.

But combination of two cells in *vertical direction* needs the following special care:

- ROW0 in the character ROM is for use in the "north-west shadowing" submode. If north-west shadowing submodes is intended for use by this formulated character font, ROW 0 should contain the bit pattern of ROW 18 of the font above it or a broken dot in the shadow may occur (see Figures 43 and 44).

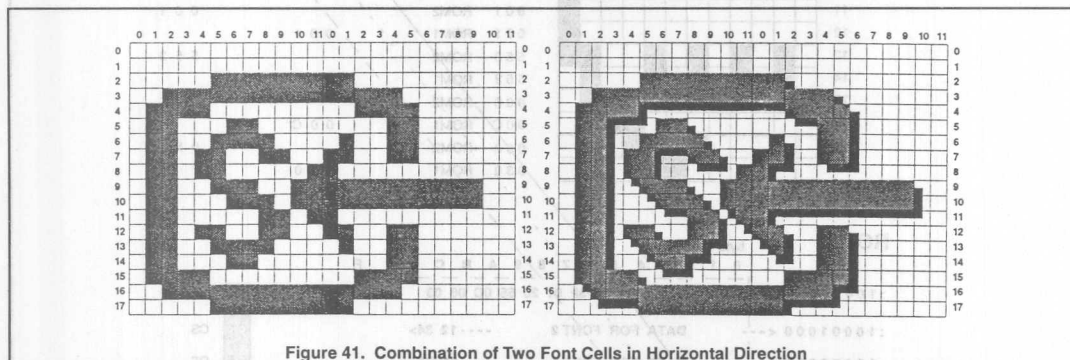


Figure 41. Combination of Two Font Cells in Horizontal Direction

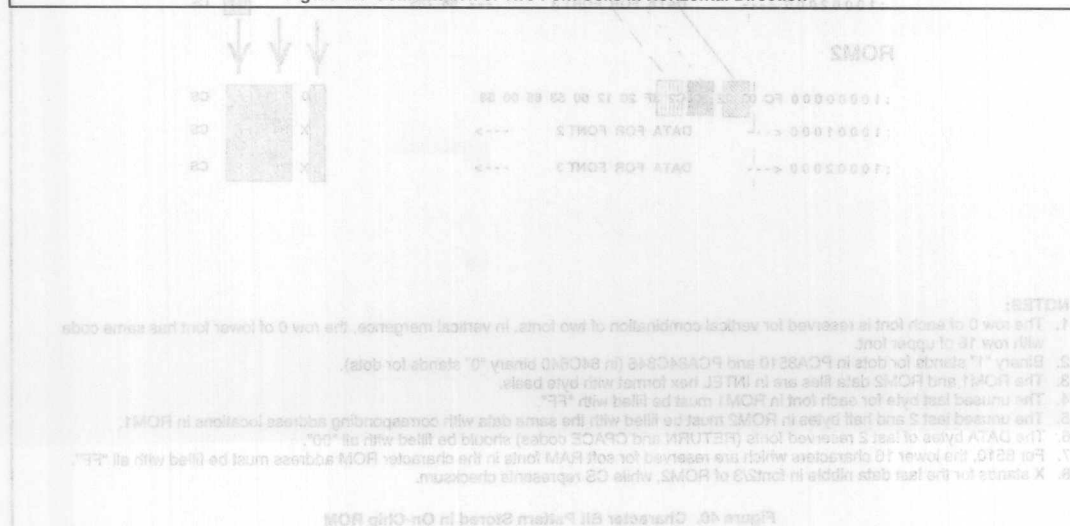
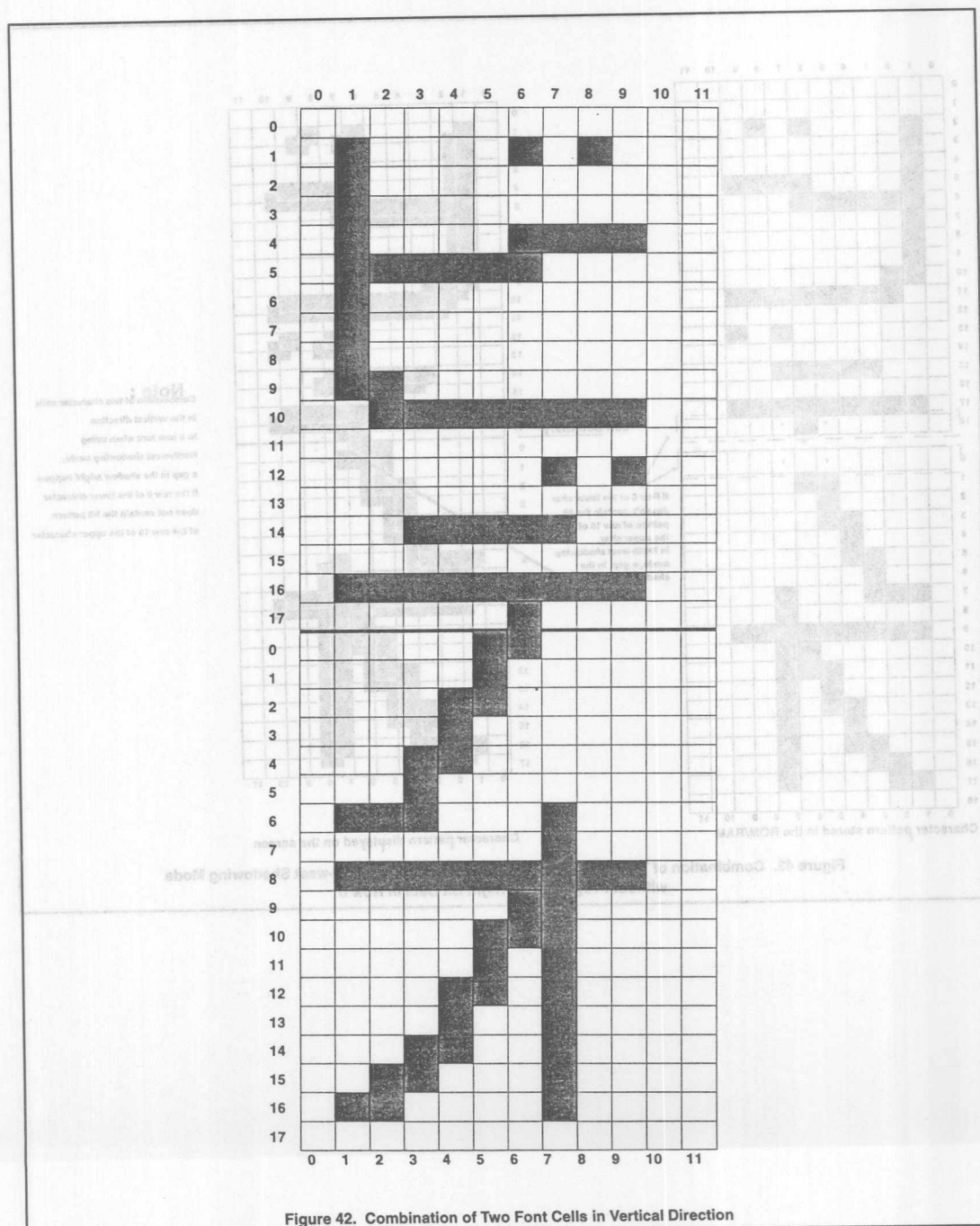


Figure 40. Character Bit Pattern Stored in On-Chip ROM



# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

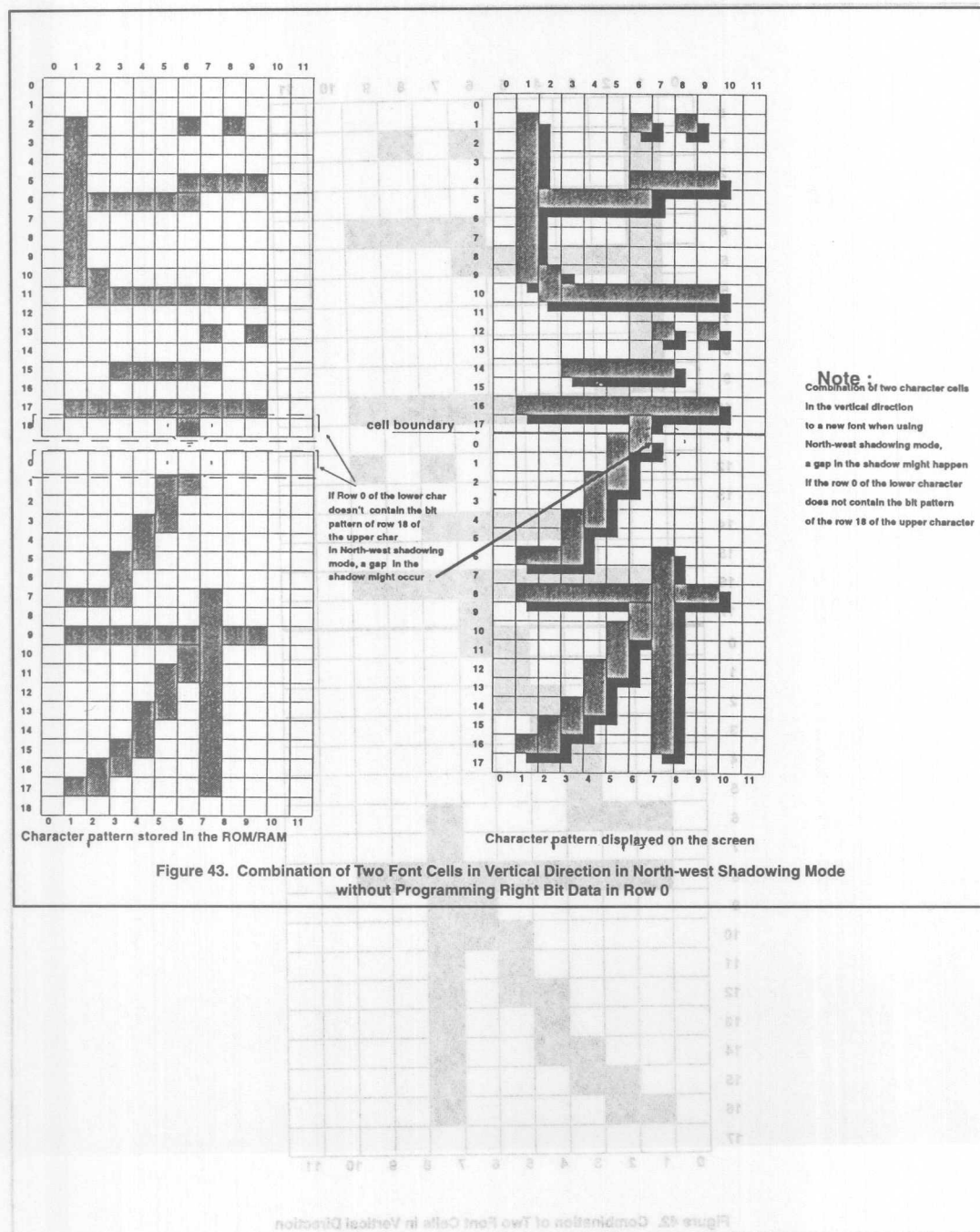
P83CL168/167/268/267





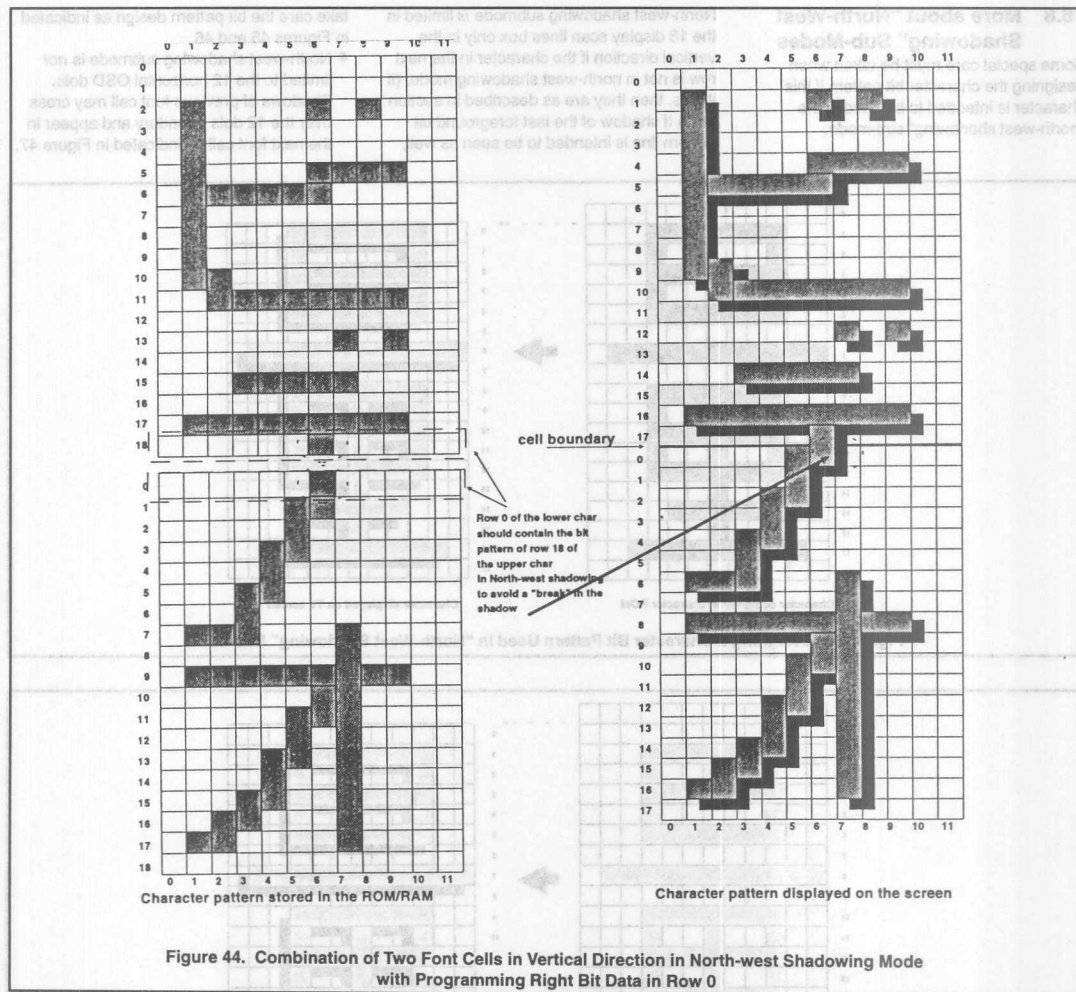
# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267



# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267



## Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

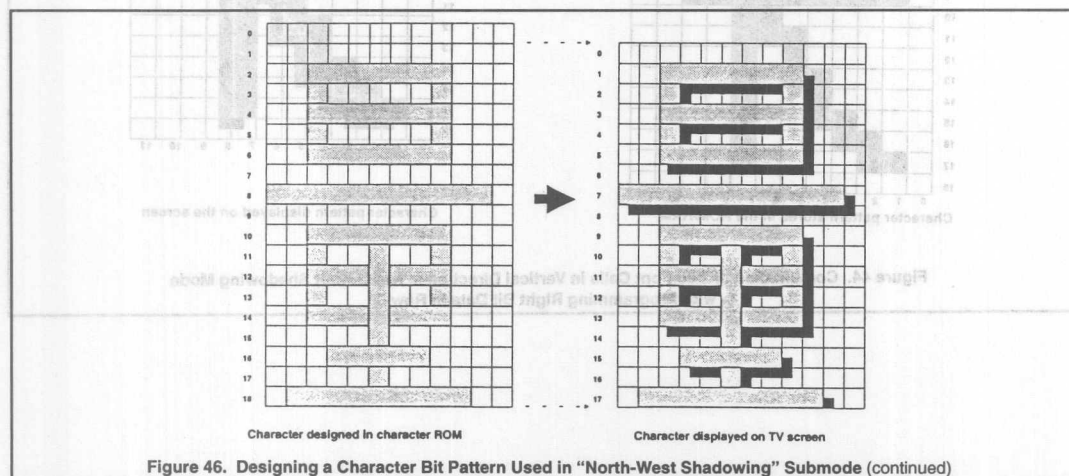
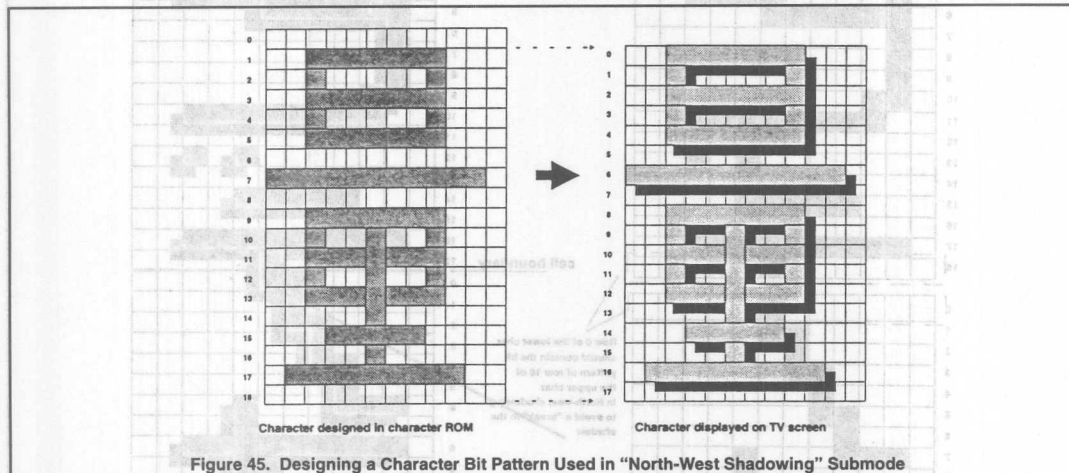
### 16.8 More about "North-West Shadowing" Sub-Modes

Some special care must be taken when designing the character bit pattern if this character is intended to be used in the "north-west shadowing" sub-mode.

North-west shadowing submode is limited in the 18 display scan lines box only in the vertical direction if the character in the next row is not in north-west shadowing mode, (if it was, then they are as described in section 16.7) If shadow of the last foreground bit pattern line is intended to be seen as well,

take care the bit pattern design as indicated in Figures 45 and 46.

- North-west shadowing submode is *not* limited to the 12 horizontal OSD dots. Shadows of previous font cell may cross over the 12 dots boundary and appear in the next font cell as indicated in Figure 47.



# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

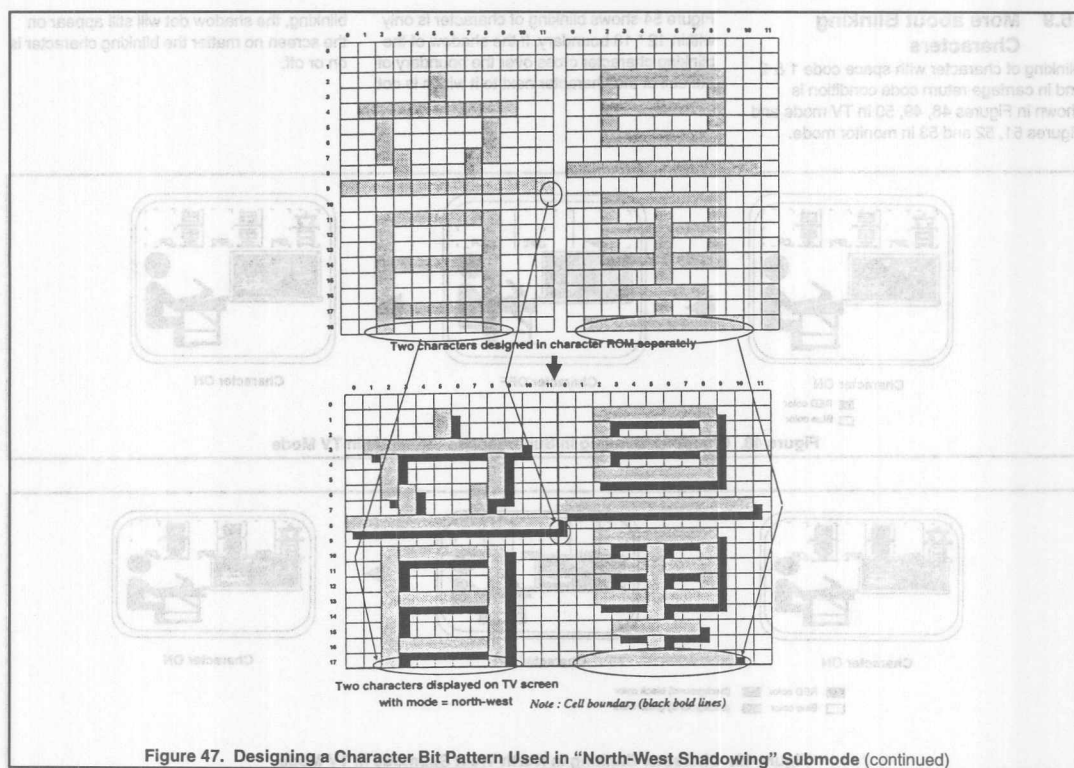
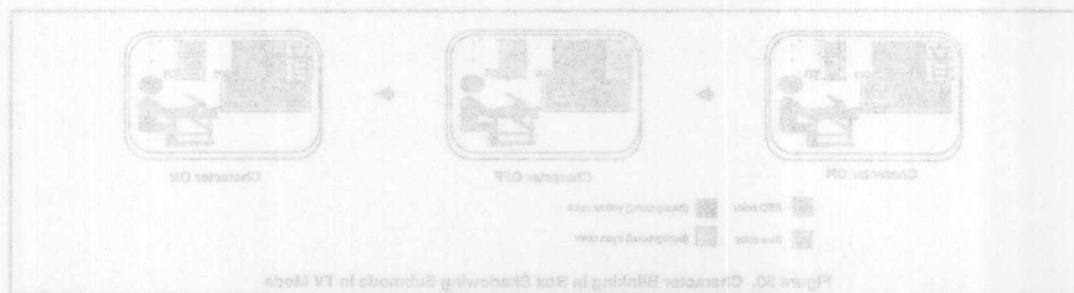


Figure 47. Designing a Character Bit Pattern Used in "North-West Shadowing" Submode (continued)



## Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

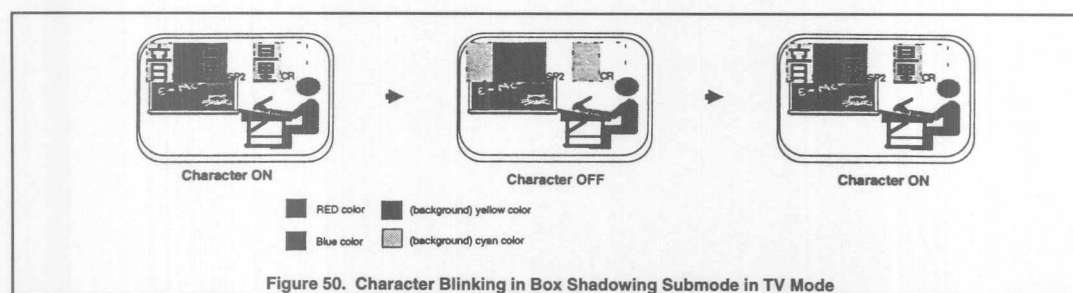
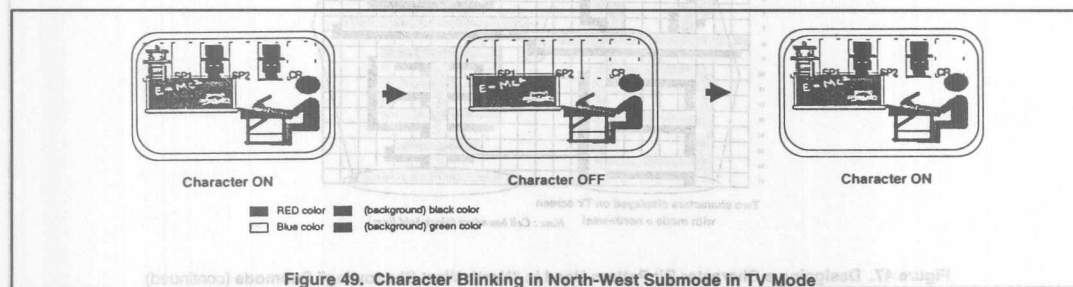
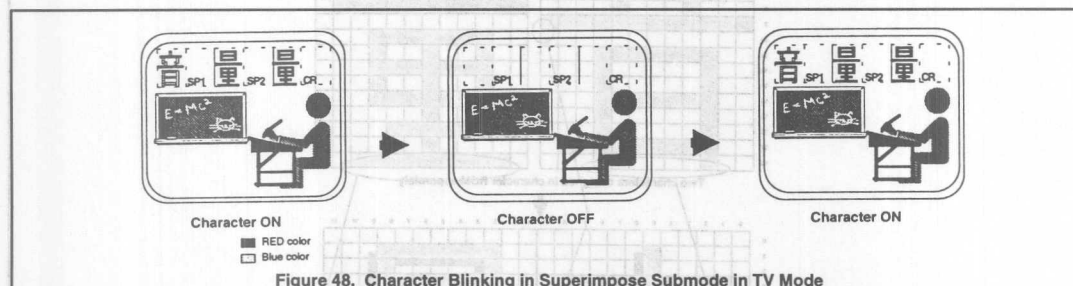
P83CL168/167/268/267

### 16.9 More about Blinking Characters

Blinking of character with space code 1 & 2 and in carriage return code condition is shown in Figures 48, 49, 50 in TV mode and Figures 51, 52 and 53 in monitor mode.

Figure 54 shows blinking of character is only within 12 \* 18 boundary. If the shadow of the blinking character cross over the boundary of the cell of the character next to it which is not

blinking, the shadow dot will still appear on the screen no matter the blinking character is on or off.





## Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

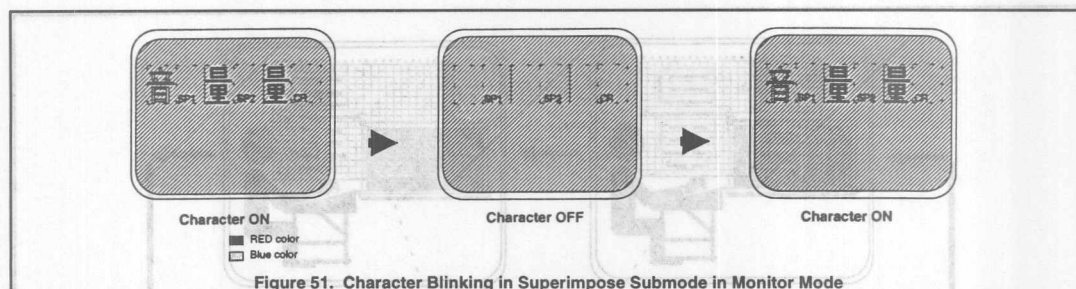


Figure 51. Character Blinking in Superimpose Submode in Monitor Mode

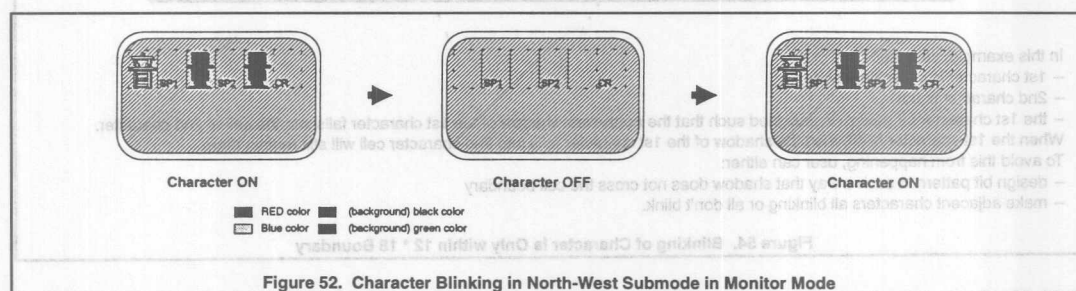


Figure 52. Character Blinking in North-West Submode in Monitor Mode

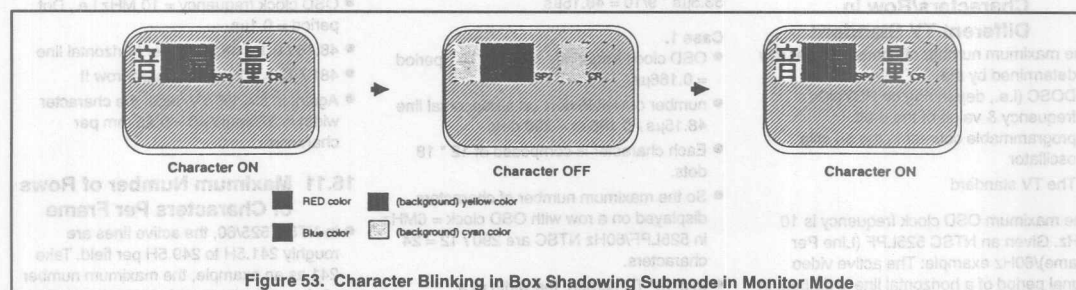
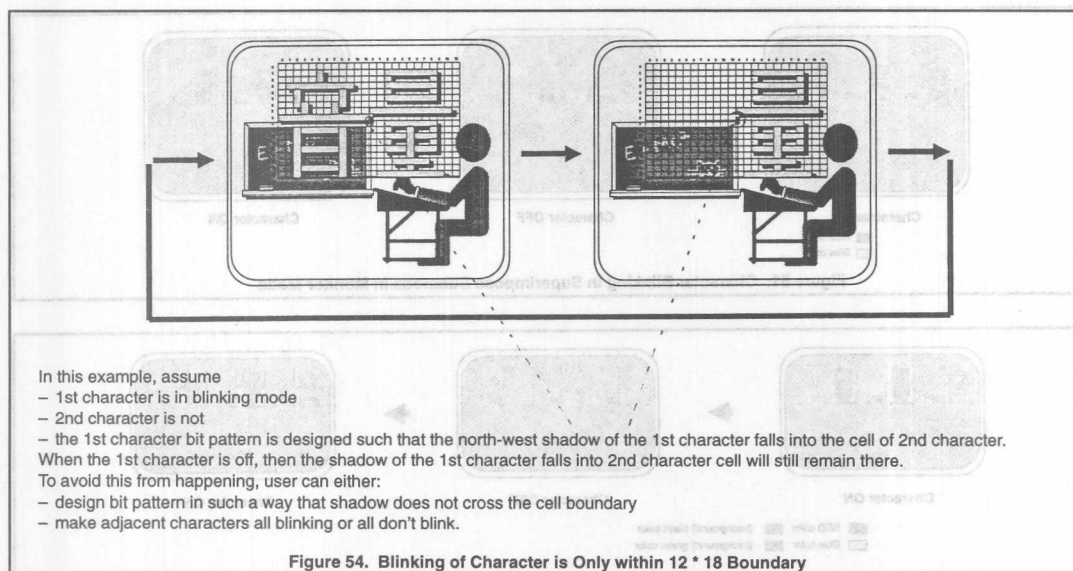


Figure 53. Character Blinking in Box Shadowing Submode in Monitor Mode



# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

## P83CL168/167/268/267



### 16.10 Maximum Number of Characters/Row in Different TV Standard

The maximum number of characters per row is determined by the

- DOSC (i.e., depending on HSYNC frequency & value of the 6-bit programmable counter in the on chip oscillator
- The TV standard

The maximum OSD clock frequency is 10 MHz. Given an NTSC 525LPF (Line Per Frame)/60Hz example: The active video signal period of a horizontal line is 53.5µs. In order to reduce the jittering of the screen edge, overscan is normally applied by the TV manufacturer. It causes the real visible video

period of a horizontal line reduced to  $53.5\mu s * 9/10 = 48.15\mu s$ .

#### Case 1.

- OSD clock frequency = 6MHz, i.e., period = 0.166µs,
- number of visible dots on a horizontal line  $48.15\mu s / 0.166\mu s = 290$  dots
- Each character is composed of 12 \* 18 dots.
- So the maximum number of characters displayed on a row with OSD clock = 6MHz in 525LPF/60Hz NTSC are  $290 / 12 = 24$  characters.
- In a 19" TV screen, the width of a horizontal line is around 370mm which gives approximately 15.4 mm/character in width.

#### Case 2.

- OSD clock frequency = 10 MHz i.e., Dot period = 0.1µs
- $48.15 / 0.1 = 481$  dots per horizontal line
- $481 / 12 = 40$  characters per row !!
- Again, in the 19" TV case, the character width is  $370\text{mm} / 40 = 9.25$  mm per character.

### 16.11 Maximum Number of Rows of Characters Per Frame

- In NTSC 525/60, the active lines are roughly 241.5H to 249.5H per field. Take 241 as an example, the maximum number of character rows per frame is (remember a character is 12 \* 18 dots):  $241 / 18 = 13$  rows.
- In PAL 625/50, it is around  $280 / 18 = 15$  rows.

## Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

### 16.12 OSD Jitter Cancelling

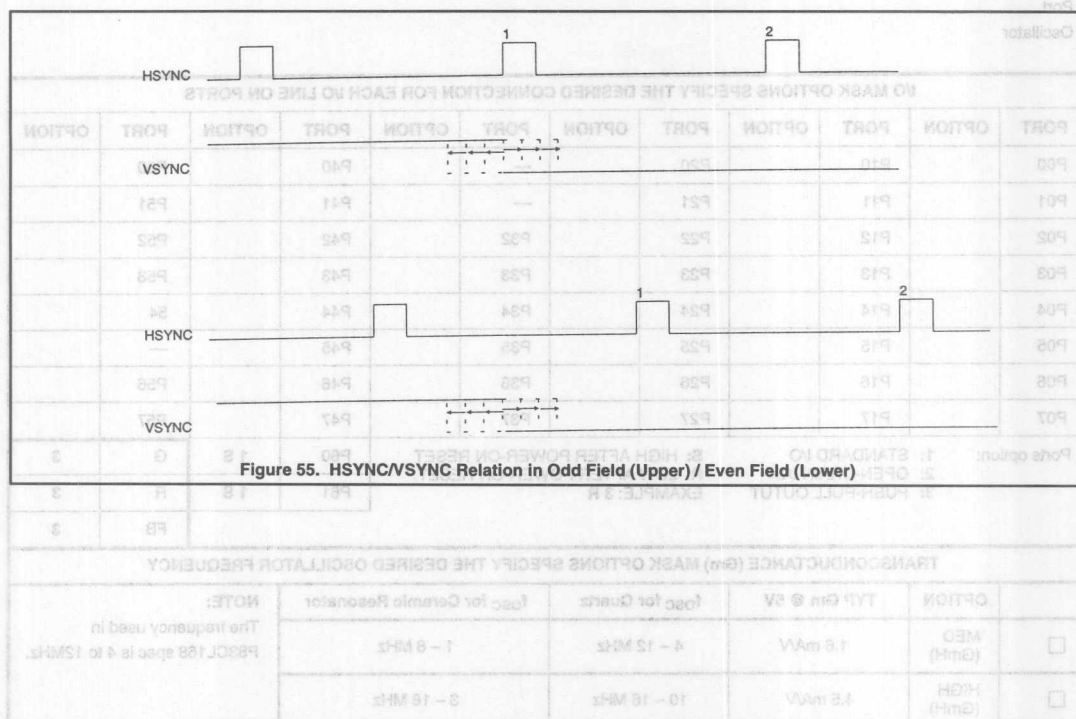
#### Circuit

The VSYNC/HSYNC relationship in normal broadcasting signals is illustrated as follow in even and odd field.

But in the condition where no broadcast signals or unstable VSYNC (e.g., play back of aged video tape) the *trailing edge* of VSYNC signal may drift forwards and backwards. Miscounting of one horizontal

scan line might happen thus character displayed on screen may drift up and down.

To avoid this happening an internal OSD jitter cancelling circuit is implemented without user's special attention.



Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

17. OPTION LIST AND SFR ADDRESS MAP SUMMARY

17.1 Option List

Options available in P83CL168 are mainly in two area,

- Port
- Oscillator

I/O MASK OPTIONS SPECIFY THE DESIRED CONNECTION FOR EACH I/O LINE ON PORTS											
PORT	OPTION	PORT	OPTION	PORT	OPTION	PORT	OPTION	PORT	OPTION	PORT	OPTION
P00		P10		P20		—		P40		P50	
P01		P11		P21		—		P41		P51	
P02		P12		P22		P32		P42		P52	
P03		P13		P23		P33		P43		P53	
P04		P14		P24		P34		P44		54	
P05		P15		P25		P35		P45		—	
P06		P16		P26		P36		P46		P56	
P07		P17		P27		P37		P47		P57	
Ports option: 1: STANDARD I/O 2: OPEN-DRAIN I/O 3: PUSH-PULL OUTUT  S: HIGH AFTER POWER-ON RESET R: LOW AFTER POWER-ON RESET EXAMPLE: 3 R								P60	1 S	G	3
								P61	1 S	R	3
										FB	3
TRANSCONDUCTANCE (Gm) MASK OPTIONS SPECIFY THE DESIRED OSCILLATOR FREQUENCY											
	OPTION	TYP Gm @ 5V	f <sub>osc</sub> for Quartz	f <sub>osc</sub> for Ceramic Resonator		NOTE: The frequency used in P83CL168 spec is 4 to 12MHz.					
<input type="checkbox"/>	MED (GmH)	1.6 mA/V	4 – 12 MHz	1 – 6 MHz							
<input type="checkbox"/>	HIGH (GmH)	4.5 mA/V	10 – 16 MHz	3 – 16 MHz							

# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

P83CL168/167/268/267

## 17.2 SFR Address Map

### Summary

This is just a summary of all the SFR in ascending address sequence for designer and software programmer's easy reference. The registers which are standard 80C410's registers are marked by \*\*\*.

### SFR ADDRESS MAP SUMMARY

Address	Names	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
80H	P0 latch	P07	P06	P05	P04	P03	P02	P01	P00
81H	SP (stack pointer)	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
82H	DPL (data pointer L)	DPL7	DPL6	DPL5	DPL4	DPL3	DPL2	DPL1	DPL0
83H	DPH (data pointer H)	DPH7	DPH6	DPH5	DPH4	DPH3	DPH2	DPH1	DPH0
86H	I2CCON (I <sup>2</sup> C port control)	—	—	—	—	—	—	SW1	SW2
87H	PCON (power control)	—	—	—	—	GF1	GF0	PD	IDL
88H	TCON (timer control)	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
89H	TMOD (timer mode)	Gate	C/T	M1	M0	Gate	C/T	M1	M0
8AH	TL0 (timer 0 LSB)	TL07	TL06	TL05	TL04	TL03	TL02	TL01	TL00
8BH	TL1 (timer 1 LSB)	TL17	TL16	TL15	TL14	TL13	TL12	TL11	TL10
8CH	TH0 (timer 0 MSB)	TH07	TH06	TH05	TH04	TH03	TH02	TH01	TH00
8DH	TH1 (timer 1 MSB)	TH17	TH16	TH15	TH14	TH13	TH12	TH11	TH10
8EH	T2 (timer/counter 2)	T27	T26	T25	T24	T23	T22	T21	T20
90H	P1 latch	P17	P16	P15	P14	P13	P12	P11	P10
98H	OSAT (osd attribute)	—	—	—	AT4	AT3	AT2	AT1	AT0
99H	OSDT (osd character data)	C7	C6	C5	C4	C3	C2	C1	C0
9AH	OSAD (osd address)	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
A0H	P2 latch	P27	P26	P25	P24	P23	P22	P21	P20
A8H	IEN0 (interrupt enable 0)	EA	—	ES1	—	ET1	EX1	ET0	EX0
B0H	P3 latch	P37	P36	P35	P34	P33	P32	—	P30
B1H	P4 latch	P47	P46	P45	P44	P43	P42	P41	P40
B2H	P5 latch	P57	P56	—	P54	P53	P52	P51	P50
B3H	P6 latch	—	—	—	—	—	—	P61	P60
B8H	IP0 (interrupt priority 0)	—	—	PS1	—	PT1	PX1	PT0	PX0
C0H	IRQ1 (interrupt request 1)	IQ9	IQ8	IQ7	IQ6	IQ5	IQ4	IQ3	IQ2
C1H	OSCON (osd control)	—	—	Mode	Hp	Vp	Bp	BF	OSDE
C2H	OSORGV (osd v start)	—	—	VP5	VP4	VP3	VP2	VP1	VP0
C3H	OSORGH (osd h start)	—	—	HP5	HP4	HP3	HP2	HP1	HP0
C4H	OSPLL (osd pll divider)	—	—	PLL5	PLL4	PLL3	PLL2	PLL1	PLL0
C5H	OSSTART (display start)	START7	START6	START5	START4	START3	START2	START1	START0
C6H	HIOD (high speed i/o data)	D7	D6	D5	D4	D3	D2	D1	D0
C7H	HIOCON (HIO control)	—	—	—	CLKE	DOUTE	CLK1	CLK0	DIR
C8H	AFCON (AFC control)	—	AFCH1	AFCH0	AFCL3	AFCL2	AFCL1	AFCL0	AFCC
C9H	VINT (Vsync interrupt)	—	—	—	—	—	—	—	VVLV

# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

## P83CL168/167/268/267

### SFR ADDRESS MAP SUMMARY (Continued)

Address	Names	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CAH	RCM (R/C preprocessor)	—	RCPOL	RCF1	RCF0	RCS1	RCS0	RCM	OVR
CBH	RCMV (R/C counter value)	LVL/D7	D6	D5	D4	D3	D2	D1	D0
CCH	P1SEL (port 1 selection)	PWM7E	PWM6E	PWM5E	PWM4E	PWM3E	PWM2E	PWM1E	PWM0E
CDH	P4SEL (port 4 selection)	AFC3E	AFC2E	AFC1E	AFC0E	TDACE	DINE	CLKE	DOUTE
CEH	P5SEL (port 5 selection)	RMCE	T2E	—	ISDBE	SCL2E	SDA2E	SCL1E	SDA1E
CFH	OSCON2 (osd bgnd color)	—	—	—	—	R	G	B	—
D0H	PSW (program status word)	CY	AC	F0	RS1	RS0	OV	—	P
D2H	VSTL (14-bit pwm low)	—	VST6	VST5	VST4	VST3	VST2	VST1	VST0
D3H	VSTH (14-bit pwm high)	—	VSTD	VSTC	VSTB	VSTA	VST9	VST8	VST7
D8H	S1CON (I <sup>2</sup> C control)	—	ENS1	STA	STO	SI	AA	CR1	CR0
D9H	S1STA (I <sup>2</sup> C status)	SC4	SC3	SC2	SC1	SC0	0	0	0
DAH	S1DAT (I <sup>2</sup> C data)	D7	D6	D5	D4	D3	D2	D1	D0
DBH	S1ADR (I <sup>2</sup> C address)	SLA6	SLA5	SLA4	SLA3	SLA2	SLA1	SLA0	GC
EOH	ACC (Accumulator)	ACC7	ACC6	ACC5	ACC4	ACC3	ACC2	ACC1	ACC0
E4H	PWM0 (7-bit PWM)	—	data6	data5	data4	data3	data2	data1	data0
E5H	PWM1 (7-bit PWM)	—	data6	data5	data4	data3	data2	data1	data0
E6H	PWM2 (7-bit PWM)	—	data6	data5	data4	data3	data2	data1	data0
E7H	PWM3 (7-bit PWM)	—	data6	data5	data4	data3	data2	data1	data0
E8H	IEN1 (interrupt enable 1)	EX9	EX8	EX7	EX6	EX5	EX4	EX3	EX2
E9H	IX1 (interrupt polarity 1)	IL9	1	1	1	1	IL4	IL3	IL2
ECH	PWM4 (6-bit PWM)	—	—	data5	data4	data3	data2	data1	data0
EDH	PWM5 (6-bit PWM)	—	—	data5	data4	data3	data2	data1	data0
EEH	PWM6 (6-bit PWM)	—	—	data5	data4	data3	data2	data1	data0
EFH	PWM7 (6-bit PWM)	—	—	data5	data4	data3	data2	data1	data0
FOH	B (B register)	B7	B6	B5	B4	B3	B2	B1	B0
F8H	IP1 (interrupt priority 1)	PX9	PX8	PX7	PX6	PX5	PX4	PX3	PX2



# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

## P83CL168/167/268/267

### APPENDIX:

#### AC/DC CHARACTERISTICS OF P83CL168 (V2.1)

MAXIMUM RATING					
SYMBOL	PARAMETER	MIN	MAX	UNIT	
V <sub>DD</sub>	Supply voltage	-0.5	+6.5	V	
V <sub>I</sub>	All input voltage	-0.5	V <sub>DD</sub> +0.5	V	
P <sub>TOT</sub>	Total power dissipation	-	1	W	
-I <sub>OH</sub>	Maximum source current for all port lines	-	10	mA	
I <sub>OL</sub>	Maximum sink current for all port lines	-	30	mA	
T <sub>STG</sub>	Storage temperature	-55	+125	°C	
T <sub>AMB</sub>	Ambient operating temperature range	-20	+70	°C	
T <sub>AMB</sub>	Operating junction temperature range	-	+125	°C	
CHARACTERISTICS					
SYMBOL	PARAMETER	MIN	TYP	MAX	UNIT
Oscillator (V <sub>DD</sub> =4.5 – 5.5V, temperature = -20°C to +70°C)					
f <sub>XTAL</sub>	Crystal frequency (V <sub>DD</sub> = 4.5 – 5.5V)	4	-	12.0	MHz
DOSC	On screen display clock (external capacitance at pin "C" = 0.1μF)	4.0	8.0	10.0	MHz
AFC (V <sub>DD</sub> =4.5 – 5.5V, temperature = -20°C to +70°C)					
T <sub>afc</sub>	Conversion time (from any change in the AFC : channel number, voltage level, enable/disable) with F <sub>xat</sub> =12MHz	6	-	8	μS
±I <sub>IL</sub>	P44/AFC0, P45/AFC1, P46/AFC2, P47/AFC13, comparator analog input input leakage	-	-	1	μA
V <sub>AI</sub>	input voltage	V <sub>SS</sub>	-	V <sub>DD</sub>	V
V <sub>AE</sub>	conversion error range	-	-	±1/2	LSB



# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

## P83CL168/167/268/267

### DC CHARACTERISTICS

$V_{DD} = 4.5V - 5.5V$ ,  $V_{SS} = 0V$ ,  $T_{AMB} = -20^{\circ}C$  to  $+70^{\circ}C$ ; All voltages with respect to  $V_{SS}$ , unless otherwise specified.

SYMBOL	PARAMETER	MIN	TYP	MAX	UNIT
$V_{DD}$	Supply voltage operating	4.5	5.0	5.5	V
$I_{DD}$	Supply current operating ( $F_{doscrc}$ , $F_{doscrc} = F_{xtal}$ ) @ $V_{DD} = 5V$ , $F_{xtal} = 12MHz$ (preliminary)	—	15	25	mA
$I_{DD}$	Supply current operating ( $F_{doscrc}$ , $F_{doscrc} = stop$ ) @ $V_{DD} = 5V$ , $F_{xtal} = 12MHz$	—	10	15	mA
<b>HSYNC, VSYNC, TEST, Port P0, P1, P2, P3, P4, P5, P6 inputs (except P44, P45, P46, P47)</b>					
$V_{IL}$	input LOW voltage	0	—	$0.3 \cdot V_{DD}$	
$V_{IH}$	input HIGH voltage	$0.7 \cdot V_{DD}$	—	$V_{DD}$	
$\pm I_{IL}$	input leakage current ( $V_{SS} < V_I < V_{DD}$ ) when open drain option selected	—	—	10	$\mu A$
$-I_{IL}$	input current at logic 1 when other options selected ( $V_{DD}=5$ , $V_{in}=0.4V$ )	—	—	100	$\mu A$
<b>Port P0 outputs</b>					
$V_{OL}$	output LOW voltage $V_{DD}=5V$ , $I_{OL}=10mA$	—	—	0.4	V
$-I_{OH}$	pull-up output HIGH voltage, $V_{DD}=5V$ ; $V_O=0.7 \cdot V_{DD}$	40	100	—	$\mu A$ (option 1)
$-I_{OH}$	pull-up output HIGH voltage, $V_{DD}=5V$ ; $V_O=V_{SS}$	—	140	400	$\mu A$ (option 1)
$-I_{OH}$	push-pull output HIGH voltage, $V_{DD}=5V$ ; $V_O=V_{DD}-0.4V$	1.6	7.0	—	mA (option 3)
<b>Port P1, P2, P3, P4, P6 and P54, P55, P56, P57 outputs</b>					
$I_{OL}$	output LOW voltage $V_{DD}=5V$ , $V_{OL}=0.4V$	1.6	7.0	—	mA
$-I_{OH}$	pull-up output HIGH voltage, $V_{DD}=5V$ ; $V_O=0.7 \cdot V_{DD}$	40	100	—	$\mu A$ (option 1)
$-I_{OH}$	pull-up output HIGH voltage, $V_{DD}=5V$ ; $V_O=V_{SS}$	—	140	400	$\mu A$ (option 1)
$-I_{OH}$	push-pull output HIGH voltage, $V_{DD}=5V$ ; $V_O=V_{DD}-0.4V$	1.6	7.0	—	mA (option 3)
<b>R, G, FB</b>					
$I_{OL}$	output LOW voltage $V_{DD}=5V$ , $V_{OL}=0.4V$	1.6	7.0	—	mA
$-I_{OH}$	push-pull output HIGH voltage, $V_{DD}=5V$ ; $V_O=V_{DD}-0.4V$	1.6	7.0	—	mA (option 3)
<b>P50/SDA1, P51/SCL1, P52/SDA2, P53/SCL2 outputs</b>					
$I_{OL}$	output LOW voltage $V_{DD}=5V$ , $V_{OL}=0.4V$	3.0	—	—	mA
$-I_{OH}$	pull-up output HIGH voltage, $V_{DD}=5V$ ; $V_O=0.7 \cdot V_{DD}$	40	100	—	$\mu A$ (option 1, not as $I^2C$ )
$-I_{OH}$	pull-up output HIGH voltage, $V_{DD}=5V$ ; $V_O=V_{SS}$	—	140	400	$\mu A$ (option 1, not as $I^2C$ )
$-I_{OH}$	push-pull output HIGH voltage, $V_{DD}=5V$ ; $V_O=V_{DD}-0.4V$	1.6	7.0	—	$\mu A$ (option 3, not as $I^2C$ )
<b>RESET</b>					
$V_{IL}$	input LOW voltage	0	—	$0.3 \cdot V_{DD}$	
$V_{IH}$	input HIGH voltage	$0.7 \cdot V_{DD}$	—	$V_{DD}$	
RRST	internal reset pull down resistor	50		200	K $\Omega$

# Single-chip 8-bit microcontroller for TV on-screen display, VS tuning and control function

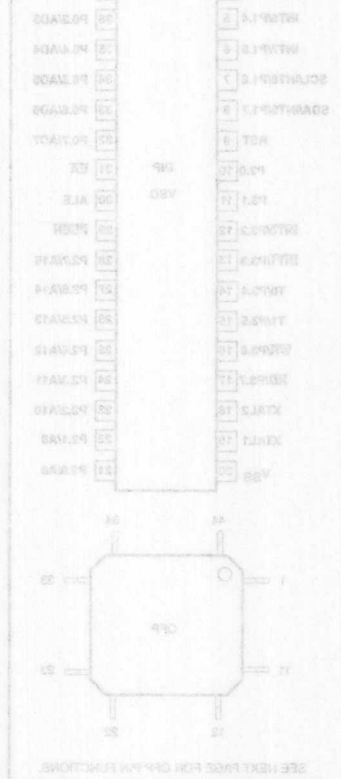
P83CL168/167/268/267

## AC CHARACTERISTICS

SYMBOL	PARAMETER	MIN	TYP	MAX	UNIT
$t_R$	P0, P1, P2, P3, P4, P5, P6 rise time (Cload=50P) <sup>1</sup>	30	—	—	ns
$t_F$	P0, P1, P2, P3, P4, P5, P6 fall time (Cload=50P) <sup>1</sup>	30	—	—	ns

### NOTES:

1. The pins related to OSD outputs, i.e., R, G, P54/B, P55/I and FB, are not slope controlled.



SEE NEXT PAGE FOR CPU PIN FUNCTIONS.

- Fully static 80C51 CPU
- 80C51 serial interface
- Two power control modes:
  - Idle mode
  - Power-down mode — can be terminated by reset or external interrupt
- Wake-up via external interrupt at port 1
- Single supply voltage 1.8V to 5.0V
- Frequency range of DC to 15MHz
- On-chip oscillator (quartz crystal, ceramic resonator, RC, LC)
- Very low power consumption
- Operating temperature range: —40 to +85°C

The 80C51 has two reduced power modes that are the same as those on the standard 80C51. The special reduced power feature of this part is that it can be stopped and then resumed operation from an external clock source. The clock can be stopped and start a period of time resumed. The 80C51 will resume operation from where it was when the code stopped with no loss of internal state RAM contents or Special Function Register contents. If the internal oscillator is used the part cannot be stopped and started, but the power-down mode, which can be terminated via an interrupt, can be used to achieve similar power savings and then resume without loss of on-chip RAM and Special Function Register values.

## ORDERING CODE

PHILIPS PART ORDER NUMBER	PART ORDER NUMBER		TEMPERATURE °C	FREQUENCY	Drawing Number
	ROMless	ROM			
P83CL168/167	P83CL168/167	P83CL168/167	—40 to +85	33MHz to 15MHz	SOT128
P83CL168/167	P83CL168/167	P83CL168/167	—40 to +85	33MHz to 15MHz	SOT168A
P83CL168/167	P83CL168/167	P83CL168/167	—40 to +85	33MHz to 15MHz	SOT167B

### NOTE:

1. Parts ordered by the Philips North America part number will be marked with the Philips part marking.

For emulation purposes, the P83CL000 (16ppack version) with 256 bytes of RAM is recommended.

# Low voltage/low power single-chip 8-bit microcontroller with I<sup>2</sup>C

## 80CL410/83CL410

### DESCRIPTION

The 80CL410/83CL410 (hereafter generically referred to as 8XCL410) is manufactured in an advanced CMOS process that allows the part to operate at supply voltages down to 1.8V and oscillator frequencies down to DC. The 8XCL410 has the same instruction set as the 80C51.

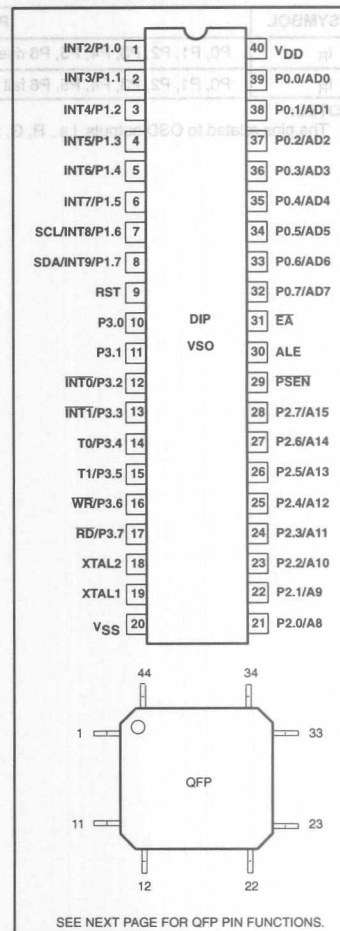
The 8XCL410 features a 4k byte ROM (83CL410), 128 bytes RAM (both ROM and RAM are externally expandable to 64k bytes), four 8-bit ports, two 16-bit timer/counters, an I<sup>2</sup>C serial interface, a thirteen source, two priority level nested interrupt structure, and on-chip oscillator circuitry suitable for quartz crystal, ceramic resonator, RC, or LC.

The 8XCL410 has two reduced power modes that are the same as those on the standard 80C51. The special reduced power feature of this part is that it can be stopped and then restarted. Running from an external clock source, the clock can be stopped and after a period of time restarted. The 8XCL410 will resume operation from where it was when the code stopped with no loss of internal state, RAM contents, or Special Function Register contents. If the internal oscillator is used the part cannot be stopped and started, but the power-down mode, which can be terminated via an interrupt, can be used to achieve similar power savings and then restart without loss of on-chip RAM and Special Function Register values.

### FEATURES

- Single supply voltage 1.8V to 6.0V
- Frequency from DC to 12MHz
- 80C51 based architecture
  - 4k × 8 ROM (64k external)
  - 128 × 8 RAM (64k external)
  - Four 8-bit I/O ports
  - Two 16-bit timer/counters
  - A thirteen-source, two-level, nested priority interrupt structure
  - 10 external interrupts
- Fully static 80C51 CPU
- I<sup>2</sup>C Serial Interface
- Two power control modes
  - Idle mode
  - Power-down mode – can be terminated by reset or external interrupt
- Wake-up via external interrupts at port 1
- Single supply voltage 1.8V to 6.0V
- Frequency range of DC to 12MHz
- On-chip oscillator (quartz crystal, ceramic resonator, RC, LC)
- Very low power consumption
- Operating temperature range:
  - 40 to +85°C

### PIN CONFIGURATION



### ORDERING CODE

PHILIPS PART ORDER NUMBER PART MARKING		PHILIPS NORTH AMERICA PART ORDER NUMBER <sup>1</sup>		TEMPERATURE °C AND PACKAGE	FREQUENCY	Drawing Number
ROMless	ROM	ROMless	ROM			
P80CL410HFP	P83CL410HFP	P80CL410HF N	P83CL410HF N	–40 to +85, 40-Pin Plastic Dual In-line Package	32kHz to 12MHz	SOT129
P80CL410HFT	P83CL410HFT	P80CL410HF D	P83CL410HF D	–40 to +85, 40-Pin Plastic Very Small Outline Package	32kHz to 12MHz	SOT158A
	P83CL410HFH			–40 to +85, 44-Pin Plastic Quad Flat Pack	32kHz to 12MHz	SOT307B

#### NOTE:

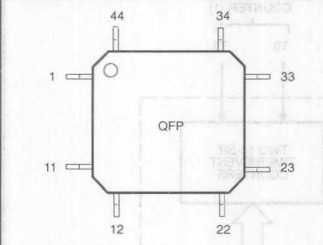
1. Parts ordered by the Philips North America part number will be marked with the Philips part marking.

For emulation purposes, the P85CL000 (Piggyback version) with 256 bytes of RAM is recommended.

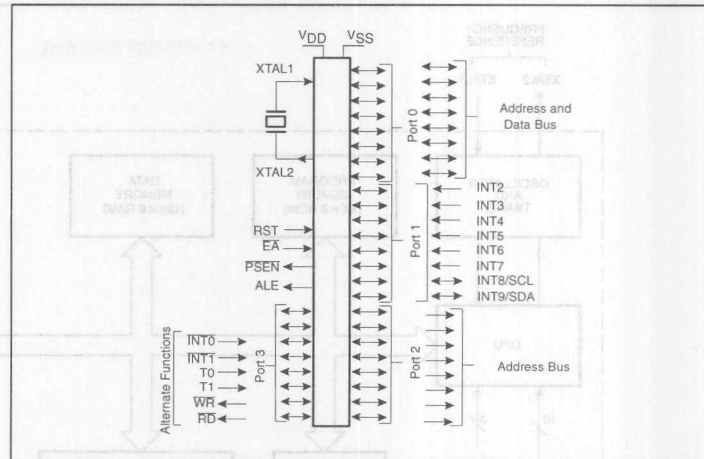
# Low voltage/low power single-chip 8-bit microcontroller with I<sup>2</sup>C

## 80CL410/83CL410

### PLASTIC QUAD FLAT PACK PIN FUNCTIONS

			
Pin	Function	Pin	Function
1	P1.5/INT7	23	P2.5/A13
2	P1.6/INT8/SCL	24	P2.6/A14
3	P1.7/INT9/SDA	25	P2.7/A15
4	RST	26	PSEN
5	P3.0	27	ALE
6	NC	28	NC
7	P3.1	29	E $\bar{A}$
8	P3.2/INT0	30	P0.7/AD7
9	P3.3/INT1	31	P0.6/AD6
10	P3.4/T0	32	P0.5/AD5
11	P3.5/T1	33	P0.4/AD4
12	P3.6/WR	34	P0.3/AD3
13	P3.7/RD	35	P0.2/AD2
14	XTAL2	36	P0.1/AD1
15	XTAL1	37	P0.0/AD0
16	VSS	38	VDD
17	NC	39	NC
18	P2.0/A8	40	P1.0/INT2
19	P2.1/A9	41	P1.1/INT3
20	P2.2/A10	42	P1.2/INT4
21	P2.3/A11	43	P1.3/INT5
22	P2.4/A12	44	P1.4/INT6

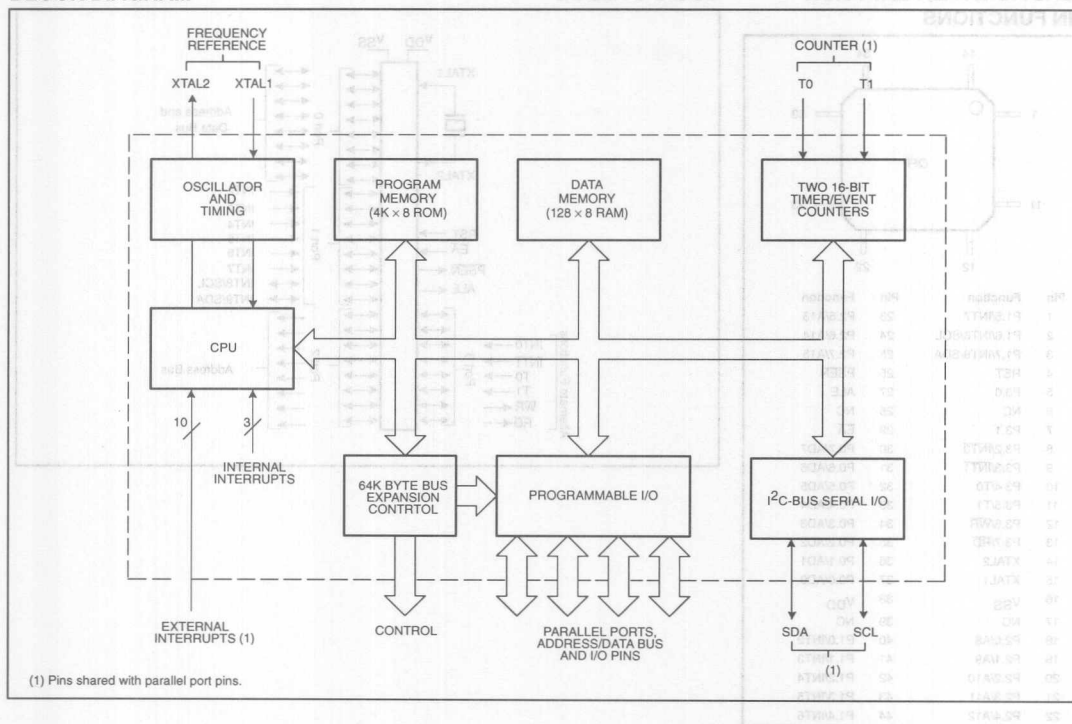
### LOGIC SYMBOL



# Low voltage/low power single-chip 8-bit microcontroller with I<sup>2</sup>C

80CL410/83CL410

## BLOCK DIAGRAM





# Low voltage/low power single-chip 8-bit microcontroller with I<sup>2</sup>C

80CL410/83CL410

## PIN DESCRIPTION

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	QFP	DIL40/ VSO40		
V <sub>SS</sub>	16	20	I	<b>Ground:</b> 0V reference.
V <sub>DD</sub>	38	40	I	<b>Power Supply:</b> This is the power supply voltage for normal, idle, and power-down operation.
P0.0–0.7	30–37	39–32	I/O	<b>Port 0:</b> Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s.
P1.0–P1.7	40–44	1–8	I/O	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Additional functions include: <b>SCL (P1.6):</b> I <sup>2</sup> C serial bus clock. <b>SDA (P1.7):</b> I <sup>2</sup> C serial bus data. <b>INT2–INT9 (P1.0–P1.7):</b> Additional external interrupts.
	1–3			
		7	I/O	
		8	I/O	
P2.0–P2.7		1–8	I	
	18–25	21–28	I/O	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register.
P3.0–P3.7	5, 7–13	10–17	I/O	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 3 also serves the special features of the 80C51 family, as listed below: <b>INT0 (P3.2):</b> External interrupt 0 <b>INT1 (P3.3):</b> External interrupt 1 <b>T0 (P3.4):</b> Timer 0 external input <b>T1 (P3.5):</b> Timer 1 external input <b>WR (P3.6):</b> External data memory write strobe <b>RD (P3.7):</b> External data memory read strobe
	8	12	I	
	9	13	I	
	10	14	I	
	11	15	I	
	12	16	O	
	13	17	O	
RST	4	9	I	<b>Reset:</b> A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to V <sub>SS</sub> permits a power-on reset using only an external capacitor to V <sub>DD</sub> .
ALE	27	30	O	<b>Address Latch Enable:</b> Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory.
PSEN	26	29	O	<b>Program Store Enable:</b> The read strobe to external program memory. When the device is executing code from the external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory. PSEN is not activated during fetches from internal program memory.
EA	29	31	I	<b>External Access Enable:</b> EA must be externally held low to enable the device to fetch code from external program memory locations 0000H to 0FFFFH. If EA is held high, the device executes from internal program memory unless the program counter contains an address greater than 0FFFFH.
XTAL1	15	19	I	<b>Crystal 1:</b> Input to the inverting oscillator amplifier and input for an external clock source.
XTAL2	14	18	O	<b>Crystal 2:</b> Output from the inverting oscillator amplifier.

Table 1. 8XCL410 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB							LSB	
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
DPTR:	Data pointer (2 bytes):										
DPH	High byte	83H									00H
DPL	Low byte	82H									00H
IP0*#	Interrupt priority 0	B8H	BF	BE	BD	BC	BB	BA	B9	B8	xx000000B
			FF	FE	FD	FC	FB	FA	F9	F8	
IP1*#	Interrupt priority 1	F8H	PX9	PX8	PX7	PX6	PX5	PX4	PX3	PX2	00H
			AF	AE	AD	AC	AB	AA	A9	A8	
IEN0*#	Interrupt enable 0	A8H	EA	—	ES1	—	ET1	EX1	ET0	EX0	00H
			EF	EE	ED	EC	EB	EA	E9	E8	
IEN1*#	Interrupt enable 1	E8H	EX9	EX8	EX7	EX6	EX5	EX4	EX3	EX2	00H
			C7	C6	C5	C4	C3	C2	C1	C0	
IRQ1*#	Interrupt request flag	C0H	IQ9	IQ8	IQ7	IQ6	IQ5	IQ4	IQ3	IQ2	00H
IX1#	Interrupt polarity	E9H									00H
PO*#	Port 0	80H	87	86	85	84	83	82	81	80	FFH
P1*	Port 1	90H	97	96	95	94	93	92	91	90	FFH
P2*	Port 2	A0H	A7	A6	A5	A4	A3	A2	A1	A0	FFH
P3*	Port 3	B0H	B7	B6	B5	B4	B3	B2	B1	B0	FFH
PCON	Power control	87H	SMOD	—	—	—	GF1	GF0	PD	IDL	0xxx0000B
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program status word	D0H	CY	AC	F0	RS1	RS0	OV	—	P	00H
S1ADR#	Slave address	DBH	DF	DE	DD	DC	DB	DA	D9	D8	00H
S1CON*#	Serial control	D8H	—	ENS1	STA	STO	SI	AA	CR1	CR0	x0000000B
S1DAT#	Serial data	DAH									00H
S1STA#	Serial status	D9H									11111000B
SP	Stack pointer	81H	8F	8E	8D	8C	8B	8A	89	88	07H
TCON*	Timer/counter control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
			GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H
TH0	Timer 0 high byte	8CH									00H
TH1	Timer 1 high byte	8DH									00H
TL0	Timer 0 low byte	8AH									00H
TL1	Timer 1 low byte	8BH									00H

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.

## Low voltage/low power single-chip 8-bit microcontroller with I<sup>2</sup>C

80CL410/83CL410

### PORT OPTIONS

The pins of port 1 (not P1.6/SCL or P1.7/SDA), port 2, and port 3 may be individually configured with one of the following port options (see Figure 1):

#### Option 1: Standard Port—

quasi-bidirectional I/O with pull-up. The strong booster pull-up p1 is turned on for two oscillator periods after a 0-to-1 transition in the port latch. See Figure 1(a).

Option 2: **Open Drain**—quasi-bidirectional I/O with n-channel open drain output. Use as an output requires the connection of an external pull-up resistor. See Figure 1(b).

Option 3: **Push-Pull**—output with drive capability in both polarities. Under this option, pins can only be used as outputs. See Figure 1(c).

The definition of port options for port 0 is slightly different.

Two cases have to be examined. First, accesses to external memory ( $\overline{EA} = 0$  or access above the built-in memory boundary), and second, I/O accesses.

### External Memory Accesses

Option 1: True 0 and 1 are written as address to the external memory (strong pull-up is used).

Option 2: An external pull-up resistor is needed for external accesses.

Option 3: Not allowed for external memory accesses as the port can only be used as output.

### I/O Accesses

Option 1: When writing a 1 to the port latch, the strong pull-up p1 will be on for two oscillator periods. No weak pull-up exists. Without an external pull-up, this option can be used as a high-impedance input.

Option 2: Open drain—quasi-bidirectional I/O with n-channel open drain output. Use as an output requires the connection of an external pull-up resistor. See Figure 1(c).

Option 3: Push-Pull—output with drive capability in both polarities. Under this option, pins can only be used as outputs.

Individual mask selection of the post-reset state is available on any of the above pins. Make your selection by appending "S" or "R" to option 1, 2, or 3 above (e.g., 1S for a standard I/O to be set after RESET or 2R for an open-drain I/O to be reset after RESET).

Option S: **Set**—after reset, this pin will be initialized High.

Option R: **Reset**—after reset, this pin will be initialized Low.

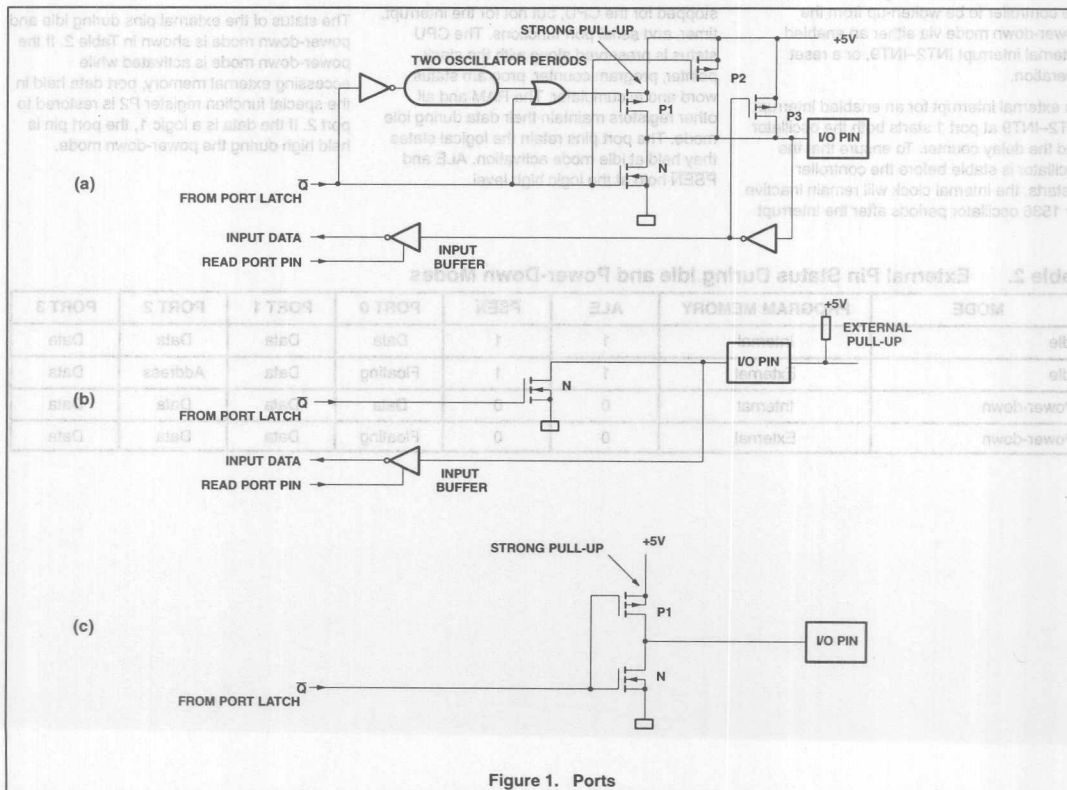


Figure 1. Ports

## Low voltage/low power single-chip 8-bit microcontroller with I<sup>2</sup>C

80CL410/83CL410

### POWER-DOWN MODE

The instruction setting PCON.1 is the last executed prior to going into the power-down mode. In power-down mode, the oscillator is stopped. The contents of the on-chip RAM and SFRs are preserved. The port pins output the values held by their respective SFRs. ALE and PSEN are held low.

In the power-down mode, V<sub>DD</sub> may be reduced to minimize power consumption. However, the supply voltage must not be reduced until the power-down mode is active, and must be restored before the hardware reset is applied and frees the oscillator. Reset must be held active until the oscillator has restarted and stabilized.

From the power-down mode the part can be restarted by using either the wake-up mode or the Reset Mode.

### Wake-Up Mode

Setting both PD and IDL bits in the PCON register forces the controller into the power-down mode. Setting both bits enable the controller to be woken-up from the power-down mode via either an enabled external interrupt INT2–INT9, or a reset operation.

An external interrupt for an enabled interrupt INT2–INT9 at port 1 starts both the oscillator and the delay counter. To ensure that the oscillator is stable before the controller restarts, the internal clock will remain inactive for 1536 oscillator periods after the interrupt

is detected. This is controlled by the on-chip delay counter. After this, the PD flag will be reset, the controller is now in the Idle mode and the interrupt will be handled in the normal way.

### Reset Mode

Setting only the PD bit in the PCON register again forces the controller into the power-down mode, but in this case it can only be restored to normal operation with a direct reset operation.

To restore normal operation, the RESET pin has to be kept High for a minimum of 24 oscillator periods. The on-chip delay counter is inactive. The user has to insure that the oscillator is stable before any operation is attempted, Figure 2 illustrates the two possibilities for wake-up.

### IDLE MODE

The instruction that sets PCON.0 is the last instruction executed before going into idle mode. In idle mode, the internal clock is stopped for the CPU, but not for the interrupt, timer, and serial port functions. The CPU status is preserved along with the stack pointer, program counter, program status word and accumulator. The RAM and all other registers maintain their data during idle mode. The port pins retain the logical states they held at idle mode activation. ALE and PSEN hold at the logic high level.

There are two methods used to terminate the idle mode. Activation of any interrupt will cause PCON to be cleared by hardware; terminating idle mode. The interrupt is serviced, and following the instruction RETI, the next instruction to be executed will be the one following the instruction that put the device in the idle mode.

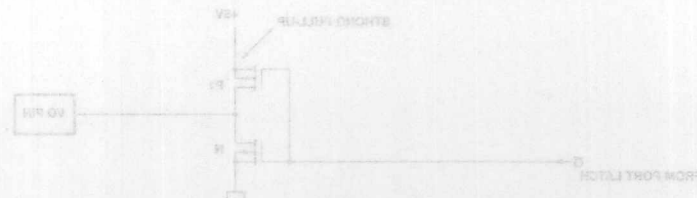
Flag bits GF0 and GF1 can be used to determine whether the interrupt was received during normal execution or idle mode. For example, the instruction that writes to PCON.0 can also set or clear one or both flag bits. When idle mode is terminated by an interrupt, the service routine can examine the status of the flag bits.

The second method of terminating the idle mode is with an external hardware reset. Since the oscillator is still running, the hardware reset is required to be active for only two machine cycles to complete the reset operation. Reset redefines all SFRs, but does not affect the state of the on-chip RAM.

The status of the external pins during idle and power-down mode is shown in Table 2. If the power-down mode is activated while accessing external memory, port data held in the special function register P2 is restored to port 2. If the data is a logic 1, the port pin is held high during the power-down mode.

Table 2. External Pin Status During Idle and Power-Down Modes

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Floating	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Floating	Data	Data	Data



# Low voltage/low power single-chip 8-bit microcontroller with I<sup>2</sup>C

80CL410/83CL410

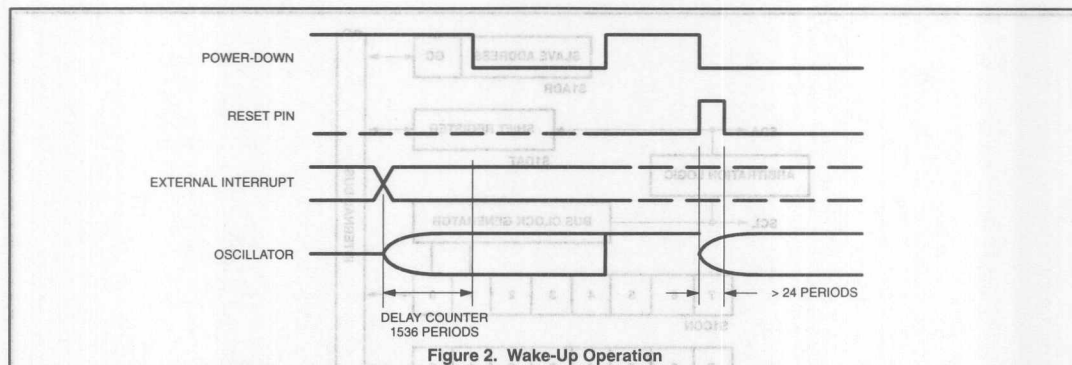


Figure 2. Wake-Up Operation

STOP flag. With this bit set while in master mode, a STOP condition is generated. When a STOP condition is detected on the bus, the SIO hardware clears the STOP flag. In the slave mode, the STOP flag may also be set to recover from an error condition. In this case, no STOP condition is transmitted to the I<sup>2</sup>C-bus. However, the SIO hardware behaves as if a STOP condition has been received and releases SDA and SCL. The SIO then switches to the "not addressed" slave receiver mode. The STOP flag is automatically cleared by hardware.

START flag. When the STA bit is set in slave mode, the SIO hardware checks the status of the I<sup>2</sup>C-bus and generates a START condition if the bus is free. If STA is set while the SIO is in master mode, SIO transmits a repeated START condition.

When ENST = 0, the SIO is disabled. The SDA and SCL outputs are in a high-impedance state; P1.6 and P1.7 function as open drain ports.

When ENST = 1, the SIO is enabled. The P1.6 and P1.7 port latches must be set to logic 1.

Assess acknowledge bit. When the AA flag is set, an acknowledge (low level to SDA) will be returned during the acknowledge clock pulse on the SCL line when:

- own slave address is received
- general call address is received (S1ADR.0 = 1)
- data byte received while device is programmed as master
- data byte received while device is selected slave

With AA = 0, no acknowledge will be returned. Consequently, no interrupt is generated when the "own slave address" or general call address is received.

SIO interrupt flag. When the SI flag is set, an acknowledge is returned after any one of the following conditions:

- a start condition is generated in master mode
- own slave address received during AA = 1
- general call address received while S1ADR.0 and AA = 1
- data byte received or transmitted in master mode (even if arbitration is lost)
- data byte received or transmitted as selected slave
- stop or start condition received as selected slave receiver or transmitter

AA

## I<sup>2</sup>C-BUS SERIAL I/O

The serial port supports the two-line I<sup>2</sup>C-bus. The I<sup>2</sup>C-bus consists of a data line (SDA) and a clock line (SCL). These lines also function as I/O port lines P1.7 and P1.6 respectively. The system is unique because data transfer, clock generation, address recognition and bus control arbitration are all controlled by hardware. The I<sup>2</sup>C-bus serial I/O has complete autonomy in byte handling and operates in four modes:

- Master transmitter
- Master receiver
- Slave transmitter
- Slave receiver

These functions are controlled by the S1CON register. S1STAT is the status register whose contents may also be used as a vector to various service routines. S1DAT is the data shift register and S1ADR the slave address register. Slave address recognition is performed by hardware.

## S1CON (D8H)

Serial control register

CRS	ENST	STA	SI	AA	CR1	CR0
-----	------	-----	----	----	-----	-----

## CR0, CR1, CR2

These three bits determine the serial clock frequency when SIO is in a master mode.



# Low voltage/low power single-chip 8-bit microcontroller with I<sup>2</sup>C

80CL410/83CL410

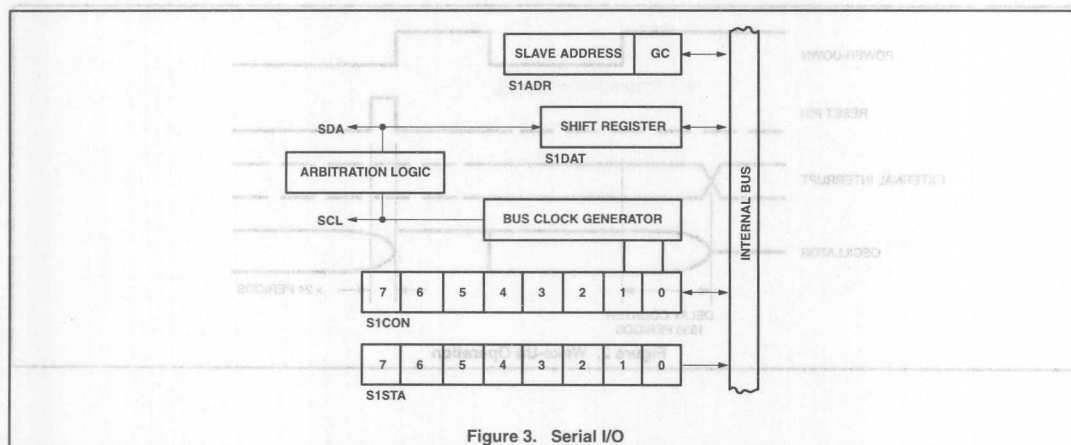


Figure 3. Serial I/O

## I<sup>2</sup>C-BUS SERIAL I/O

The serial port supports the twin line I<sup>2</sup>C-bus. The I<sup>2</sup>C-bus consists of a data line (SDA) and a clock line (SCL). These lines also function as I/O port lines P1.7 and P1.6 respectively. The system is unique because data transport, clock generation, address recognition and bus control arbitration are all controlled by hardware. The I<sup>2</sup>C-bus serial I/O has complete autonomy in byte handling and operates in four modes:

- Master transmitter
- Master receiver
- Slave transmitter
- Slave receiver

These functions are controlled by the S1CON register. S1STA is the status register whose contents may also be used as a vector to various service routines. S1DAT is the data shift register and S1ADR the slave address register. Slave address recognition is performed by hardware.

### S1CON (D8H)

Serial control register

CR2	ENS1	STA	STO	SI	AA	CR1	CR0
-----	------	-----	-----	----	----	-----	-----

### CR0, CR1, CR2

These three bits determine the serial clock frequency when SIO is in a master mode.

### AA

Assert acknowledge bit. When the AA flag is set, an acknowledge (low level to SDA) will be returned during the acknowledge clock pulse on the SCL line when:

- own slave address is received
- general call address is received (S1ADR.0 = 1)
- data byte received while device is programmed as master
- data byte received while device is selected slave

With AA = 0, no acknowledge will be returned. Consequently, no interrupt is requested when the "own slave address" or general call address is received.

### SI

SIO interrupt flag. When the SI flag is set, an acknowledge is returned after any one of the following conditions:

- a start condition is generated in master mode
- own slave address received during AA = 1
- general call address received while S1ADR.0 and AA = 1
- data byte received or transmitted in master mode (even if arbitration is lost)
- data byte received or transmitted as selected slave
- stop or start condition received as selected slave receiver or transmitter

### STO

STOP flag. With this bit set while in master mode, a STOP condition is generated. When a STOP condition is detected on the bus, the SIO hardware clears the STO flag. In the slave mode, the STO flag may also be set to recover from an error condition. In this case, no STOP condition is transmitted to the I<sup>2</sup>C-bus. However, the SIO hardware behaves as if a STOP condition has been received and releases SDA and SCL. The SIO then switches to the "not addressed" slave receiver mode. The STO flag is automatically cleared by hardware.

### STA

START flag. When the STA bit is set in slave mode, the SIO hardware checks the status of the I<sup>2</sup>C-bus and generates a START condition if the bus is free. If STA is set while the SIO is in master mode, SIO transmits a repeated START condition.

### ENS1

When ENS1 = 0, the SIO is disabled. The SDA and SCL outputs are in a high-impedance state; P1.6 and P1.7 function as open drain ports.

When ENS1 = 1, the SIO is enabled. The P1.6 and P1.7 port latches must be set to logic 1.

# Low voltage/low power single-chip 8-bit microcontroller with I<sup>2</sup>C

80CL410/83CL410

## S1STA (D9H) Status register

SC4	SC3	SC2	SC1	SC0	0	0	0	0
-----	-----	-----	-----	-----	---	---	---	---

S1STA is an 8-bit read-only special function register. S1STA.3–S1STA.7 hold a status code. S1STA.0–S1STA.2 are held LOW. The contents of S1STA may be used as a vector to a service routine. This optimizes response time of the software and consequently that of the I<sup>2</sup>C-bus.

The following is a list of the status codes:

### Abbreviations used:

SLA: 7-bit slave address

R: Read bit

W: Write bit

ACK: Acknowledgement (acknowledge bit = 0)

ACK: Not Acknowledge (acknowledge bit = 1)

DATA: 8-bit byte to or from the I<sup>2</sup>C-bus

MST: Master

SLV: Slave

TRX: Transmitter

REC: Receiver

### MST/TRX mode

S1STA value

08H – a START condition has been transmitted

10H – a repeated START condition has been transmitted

18H – SLA and W have been transmitted, ACK received

20H – SLA and W have been transmitted, ACK received

28H – DATA of S1DAT has been transmitted, ACK received

30H – DATA of S1DAT has been transmitted, ACK received

38H – Arbitration lost in SLA, R/W or DATA

### MST/REC mode

S1STA value

08H – a START condition has been transmitted

10H – a repeated START condition has been transmitted

38H – Arbitration lost while returning ACK

40H – SLA and R have been transmitted, ACK received

48H – SLA and R have been transmitted, ACK received

50H – DATA has been received, ACK returned

58H – DATA has been received, ACK returned

### SLV/REC mode

S1STA value

60H – Own SLA and W have been received, ACK returned

68H – Arbitration lost in SLA, R/W as MST. Own SLA and W have been received, ACK returned

70H – General CALL has been received, ACK returned

78H – Arbitration lost in SLA, R/W as MST. General CALL has been received

80H – Previously addressed with own SLA. DATA byte received, ACK returned

88H – Previously addressed with own SLA. DATA byte received, ACK returned

90H – Previously addressed with general CALL. DATA byte has been received, ACK has been returned

98H – Previously addressed with general CALL. DATA byte has been received, ACK has been returned

A0H – A STOP condition or repeated START condition has been received while still addressed as SLV/REC or SLV/TRX

### SLV/TRX mode

S1STA value

A8H – Own SLA and R have been received, ACK returned

B0H – Arbitration lost in SLA, R/W as MST. Own SLA and R have been received, ACK returned

B8H – DATA byte has been transmitted, ACK received

C0H – DATA byte has been transmitted, ACK received

C8H – Last DATA byte has been transmitted (AA = logic 0), ACK received

### Miscellaneous

S1STA value

00H – Bus error during MST mode or selected SLV mode, due to an erroneous START or STOP condition

F8H – No relevant state interruption available, SI = 0.

### S1DAT (DAH)

Data Shift Register

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

### Data shift register S1DAT

This register contains the serial data to be transmitted or data that has just been received. Bit 7 is transmitted or received first, i.e., data is shifted from left to right.

### S1ADR (DBH)

Slave Address Register

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

S1ADR.0, GC: 0 = general CALL address is not recognized

1 = general CALL address is recognized

S1ADR.7-1: own slave address

This 8-bit register may be loaded with the 7-bit slave address, to which the controller will respond when programmed as a slave receiver/transmitter. The LSB bit (GC) is used to determine whether the general CALL address is recognized.

Table 3. SCL Frequency

CR2	CR1	CR0	f <sub>osc</sub> DIVIDED BY	BIT RATE (kHz) at f <sub>osc</sub>		
				3.58MHz	6MHz	12MHz
0	0	0	256	14.0	23.4	46.9
0	0	1	224	16.0	26.8	53.6
0	1	0	192	18.6	31.3	62.5
0	1	1	160	22.4	37.5	75.0
1	0	0	960	3.73	6.25	12.5
1	0	1	120	29.8	50	100
1	1	0	60	59.7	100	—
1	1	1	not allowed	—	—	—

# Low voltage/low power single-chip 8-bit microcontroller with I<sup>2</sup>C

80CL410/83CL410

## INTERRUPT SYSTEM

External events and the real-time-driven on-chip peripherals require service by the CPU asynchronous to the execution of any particular section of code. To tie the asynchronous activities of these functions to normal program execution, a multiple-source, two-priority level, nested interrupt system is provided. The 8XCL410 acknowledges interrupt requests from thirteen sources, as follows:

- INT0 and INT1
- Timer 0 and timer 1
- I<sup>2</sup>C-bus serial I/O interrupt
- INT2 to INT9 (port 1)

Each interrupt vectors to a separate location in program memory for its service routine.

Each source can be individually enabled or disabled by corresponding bits in the internal enable registers (IEN0, IEN1). The priority level is selected via the interrupt priority register (IP0, IP1). All enabled sources can be globally disabled or enabled.

## External Interrupts INT2–INT9

Port 1 lines serve an alternative purpose as eight additional interrupts INT2–INT9. When enabled, each of these lines can “wake-up” the device from power-down mode. Using the IX1 register, each pin may be initialized to either active high or low. IRQ1 is the interrupt request flag register. Each flag, if the interrupt is enabled, will be set on an interrupt request but it must be cleared by software.

## IEN0 (A8H)

### Interrupt enable register

7	6	5	4	3	2	1	0
EA	—	ES1	—	ET1	EX1	ET0	EX0

Bit	Symbol	Function
IEN0.7	EEA	General enable/disable control 0 = no interrupt is enabled 1 = any individually enabled interrupt will be accepted
IEN0.6	—	(unused)
IEN0.5	ES1	Enable I <sup>2</sup> C SIO interrupt
IEN0.4	—	(unused)
IEN0.3	ET1	Enable Timer T1 interrupt
IEN0.2	EX1	Enable external interrupt 1
IEN0.1	ET0	Enable Timer T0 interrupt
IEN0.0	EX0	Enable external interrupt 0

## IEN1 (E8H)

### Interrupt enable register

7	6	5	4	3	2	1	0
EX9	EX8	EX7	EX6	EX5	EX4	EX3	EX2

Bit	Symbol	Function
IEN1.7	EX9	Enable external interrupt 9
IEN1.6	EX8	Enable external interrupt 8
IEN1.5	EX7	Enable external interrupt 7
IEN1.4	EX6	Enable external interrupt 6
IEN1.3	EX5	Enable external interrupt 5
IEN1.2	EX4	Enable external interrupt 4
IEN1.1	EX3	Enable external interrupt 3
IEN1.0	EX2	Enable external interrupt 2

where 0 = interrupt disabled  
1 = interrupt enabled

## IP0 (B8H)

### Interrupt priority register

7	6	5	4	3	2	1	0
—	—	PS1	—	PT1	PX1	PT0	PX0

Bit	Symbol	Function
IP0.7	—	(unused)
IP0.6	—	(unused)
IP0.5	PS1	I <sup>2</sup> C SIO interrupt priority level
IP0.4	—	(unused)
IP0.3	PT1	Timer 1 interrupt priority level
IP0.2	PX1	External interrupt 1 priority level
IP0.1	PT0	Timer 0 interrupt priority level
IP0.0	PX0	External interrupt 0 priority level

## IP1 (F8H)

### Interrupt priority register

7	6	5	4	3	2	1	0
PX9	PX8	PX7	PX6	PX5	PX4	PX3	PX2

Bit	Symbol	Function
IP1.7	PX9	External interrupt 9 priority level
IP1.6	PX8	External interrupt 8 priority level
IP1.5	PX7	External interrupt 7 priority level
IP1.4	PX6	External interrupt 6 priority level
IP1.3	PX5	External interrupt 5 priority level
IP1.2	PX4	External interrupt 4 priority level
IP1.1	PX3	External interrupt 3 priority level
IP1.0	PX2	External interrupt 2 priority level

Interrupt priority is as follows:  
0 – low priority  
1 – high priority

## IX1 (E9H)

### Interrupt polarity register

7	6	5	4	3	2	1	0
IL9	IL8	IL7	IL6	IL5	IL4	IL3	IL2

Bit	Symbol	Function
IX1.7	IL9	External interrupt 9 polarity level
IX1.6	IL8	External interrupt 8 polarity level
IX1.5	IL7	External interrupt 7 polarity level
IX1.4	IL6	External interrupt 6 polarity level
IX1.3	IL5	External interrupt 5 polarity level
IX1.2	IL4	External interrupt 4 polarity level
IX1.1	IL3	External interrupt 3 polarity level
IX1.0	IL2	External interrupt 2 polarity level

Writing either a “1” or “0” to an IX1 register bit sets the priority level of the corresponding external interrupt to active High or Low, respectively.

# Low voltage/low power single-chip 8-bit microcontroller with I<sup>2</sup>C

80CL410/83CL410

## IRQ1 (C0H)

### Interrupt request flag register

	7	6	5	4	3	2	1	0
	IQ9	IQ8	IQ7	IQ6	IQ5	IQ4	IQ3	IQ2
Bit	Symbol	Function						
IRQ1.7	IQ9	External interrupt 9 request flag						
IRQ1.6	IQ8	External interrupt 8 request flag						
IRQ1.5	IQ7	External interrupt 7 request flag						
IRQ1.4	IQ6	External interrupt 6 request flag						
IRQ1.3	IQ5	External interrupt 5 request flag						
IRQ1.2	IQ4	External interrupt 4 request flag						
IRQ1.1	IQ3	External interrupt 3 request flag						
IRQ1.0	IQ2	External interrupt 2 request flag						

Priority	Vector	Source
X0 (highest)	0003H	External 0
S1	002BH	I <sup>2</sup> C port
X5	0053H	External 5
T0	000BH	Timer 0
X6	005BH	External 6
X1	0013H	External 1
X2	003BH	External 2
X7	0063H	External 7
T1	001BH	Timer 1
X3	0043H	External 3
X8	006BH	External 8
X4	004BH	External 4
X9 (lowest)	0073H	External 9

Register	Function	SFR Address
IX1	Interrupt polarity register	E9H
IRQ1	Interrupt request flag register	C0H
IEN0	Interrupt enable register	A8H
IEN1	Interrupt enable register (INT2–INT9)	E8H
IP0	Interrupt priority register	B8H
IP1	Interrupt priority register (INT2–INT9)	F8H

## OSCILLATOR CIRCUITRY

The on-chip oscillator circuitry of the 8XCL410 is a single stage inverting amplifier biased by an internal feedback resistor. (See Figure 4.) The oscillator can be operated with a quartz crystal, ceramic resonator, LC network or RC network. See Figure 5 for different configurations. When ordering parts, it is necessary to specify an oscillator option. The options are: RC when an RC network will be used, OSC 2 for oscillator operation below 4MHz, OSC 3 for oscillator operation from 4MHz to 10MHz, OSC 4 for oscillator operation above 10MHz, and 32kHz if 32kHz to 400kHz operation is desired.

For operation as a standard quartz oscillator, no external components are needed (except at 32KHz). When using external capacitors, ceramic resonators, coils, and RC networks to drive the oscillator, five different configurations are supported (see Figure 5 and Table 4).

In the power-down mode the oscillator is stopped and XTAL1 is pulled high. The oscillator inverter is switched off to ensure no current will flow. To drive the device with an external clock source, apply the external clock signal to XTAL1, and leave XTAL2 to float, as shown in Figure 5(f). There are no requirements on the duty cycle of the external clock, since the input to the internal clocking circuitry is split using a flip-flop.

The following options are provided for optimum on-chip oscillator performance. Please state option when ordering:

Osc.1: Figure 5(c). An option for 32kHz clock applications with external trimmer for frequency adjustment.

A 4.7M $\Omega$  bias resistor must be connected in parallel with the crystal.

Osc.2: Figure 5(e). An option for low-power, low-frequency operations using LC components or quartz.

Osc.3: An option for medium frequency range applications.

Osc.4: An option for high frequency range applications.

RC: Figure 5(g). An option for an RC oscillator.

The equivalent circuit data of the internal oscillator compares with that of matched crystals.

The externally adjustable RC oscillator has a frequency range from 100kHz to 500kHz. (See Figure 7.)

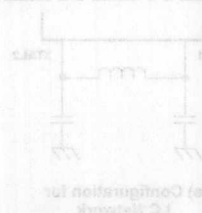
## Power-on Reset

The 8XCL410 contains on-chip circuitry which switch the port pins to the customer-defined logic level as soon as V<sub>DD</sub> exceeds 1.3V if the mask option "ON" has been chosen (see Figures 8 and 9). As soon as the minimum supply voltage is reached, the oscillator will start up. However, to ensure that the oscillator is stable before the controller starts, the clock signals are gated away from the CPU for a further 1536 oscillator periods.

An hysteresis of approximately 50mV at a typical power-on switching level of 1.3V will ensure correct operation.

The on-chip power-on reset circuitry can also be switched off via the mask option "OFF". This option reduces the power-down current to typically 800 $\mu$ A and can be chosen if external reset circuitry is used. For applications not requiring the internal reset, option "OFF" should be chosen.

An automatic reset can be obtained at power-on by connecting the RST pin to V<sub>DD</sub> via a 10 $\mu$ F capacitor. At power-on, the voltage on the RST pin is equal to V<sub>DD</sub> minus the capacitor voltage, and decreases from V<sub>DD</sub> as the capacitor discharges through the internal resistor R<sub>RST</sub> to ground. The larger the capacitor, the more slowly V<sub>RST</sub> decreases. V<sub>RST</sub> must remain above the lower threshold of the Schmitt trigger long enough to effect a complete reset. The time required is the oscillator start-up time, plus 2 machine cycles.



# Low voltage/low power single-chip 8-bit microcontroller with I<sup>2</sup>C

80CL410/83CL410

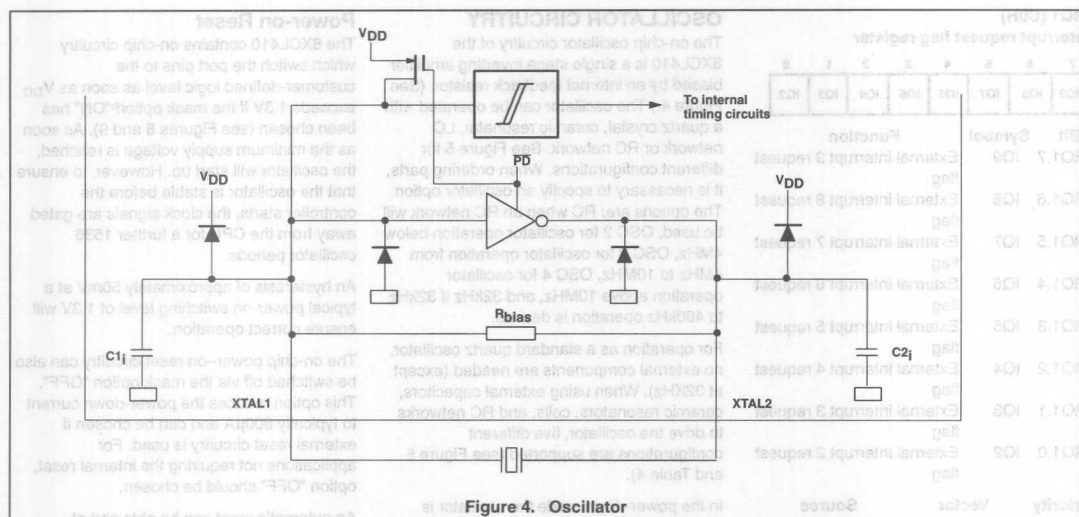


Figure 4. Oscillator

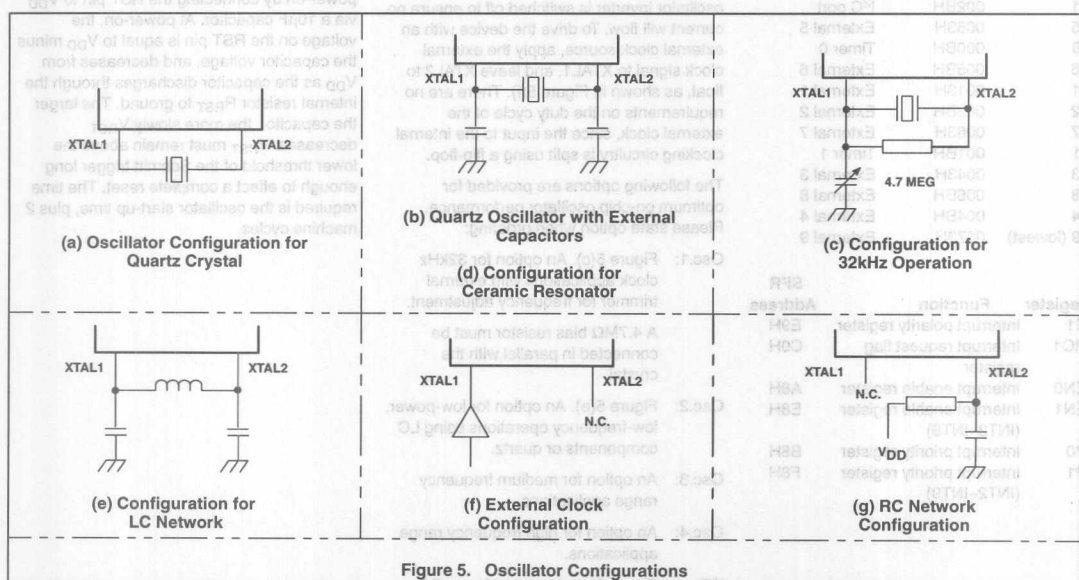


Figure 5. Oscillator Configurations



# Low voltage/low power single-chip 8-bit microcontroller with I<sup>2</sup>C

# 80CL410/83CL410

**Table 4. Oscillator Type Selection Guide**

RESONATOR	f (MHz)	OPTION	C1 EXT.		C2 EXT.		MAXIMUM RESONATOR SERIES RESISTANCE
			MIN	MAX	MIN	MAX	
Quartz	0.032	Osc.1	5	15	0	0	15kΩ <sup>1</sup>
Quartz	1.0	Osc.2	0	30	0	30	600Ω
Quartz	3.58	Osc.2	0	15	0	15	100Ω
Quartz	4.0	Osc.2	0	20	0	20	75Ω
Quartz	6.0	Osc.3	0	10	0	10	60Ω
Quartz	10.0	Osc.4	0	15	0	15	60Ω
Quartz	12.0	Osc.4	0	10	0	10	40Ω
Quartz	16.0	Osc.4	0	15	0	15	20Ω
PXE	0.455	Osc.2	40	50	40	50	10Ω
PXE	1.0	Osc.2	15	50	15	50	100Ω
PXE	3.58	Osc.2	0	40	0	40	10Ω
PXE	4.0	Osc.2	0	40	0	40	10Ω
PXE	6.0	Osc.2	0	20	0	20	5Ω
PXE	10.0	Osc.3	0	15	0	15	6Ω
PXE	12.0	Osc.4	10	40	10	40	6Ω
LC		Osc.2	20	90	20	90	10μH = 1Ω 100μH = 5Ω 1mH = 75Ω

**NOTE:**

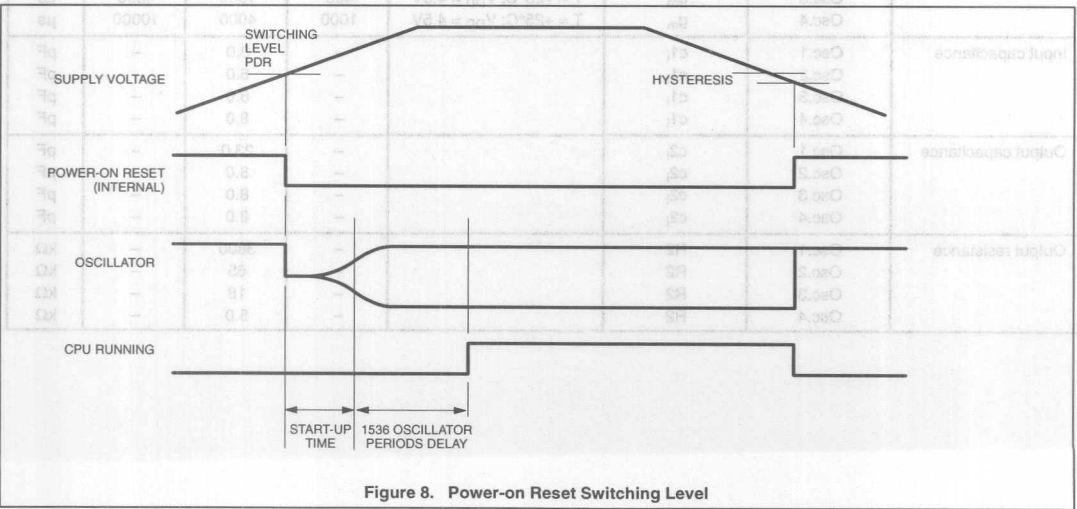
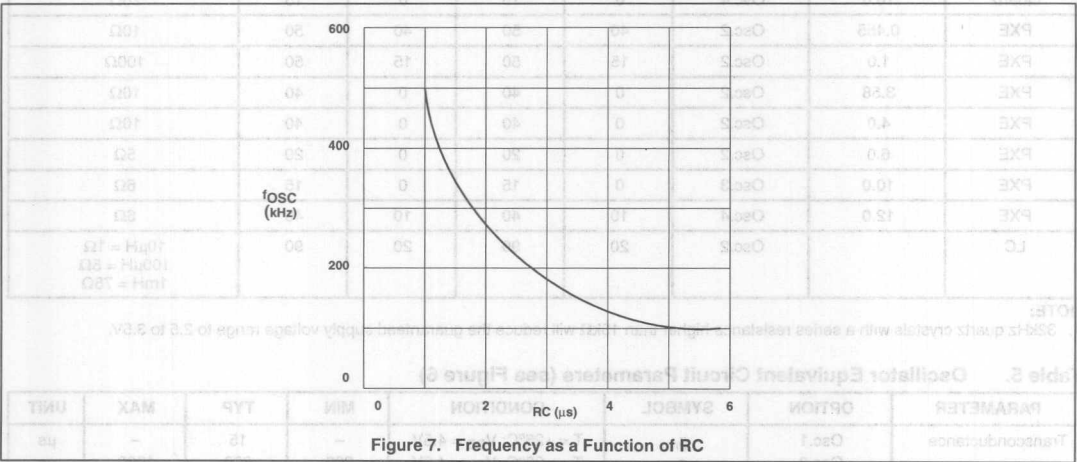
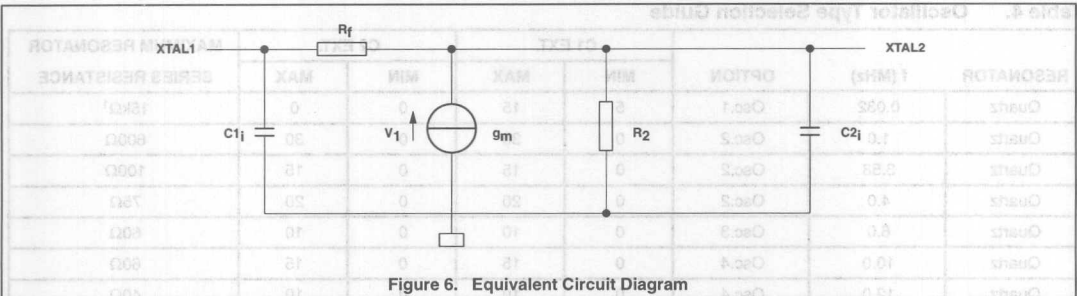
1. 32kHz quartz crystals with a series resistance higher than 15kΩ will reduce the guaranteed supply voltage range to 2.5 to 3.5V.

**Table 5. Oscillator Equivalent Circuit Parameters (see Figure 6)**

PARAMETER	OPTION	SYMBOL	CONDITION	MIN	TYP	MAX	UNIT
Transconductance	Osc.1	g <sub>m</sub>	T = +25°C; V <sub>DD</sub> = 4.5V	—	15	—	μs
	Osc.2	g <sub>m</sub>	T = +25°C; V <sub>DD</sub> = 4.5V	200	600	1000	μs
	Osc.3	g <sub>m</sub>	T = +25°C; V <sub>DD</sub> = 4.5V	400	1500	4000	μs
	Osc.4	g <sub>m</sub>	T = +25°C; V <sub>DD</sub> = 4.5V	1000	4000	10000	μs
Input capacitance	Osc.1	c <sub>1i</sub>		—	3.0	—	pF
	Osc.2	c <sub>1i</sub>		—	8.0	—	pF
	Osc.3	c <sub>1i</sub>		—	8.0	—	pF
	Osc.4	c <sub>1i</sub>		—	8.0	—	pF
Output capacitance	Osc.1	c <sub>2i</sub>		—	23.0	—	pF
	Osc.2	c <sub>2i</sub>		—	8.0	—	pF
	Osc.3	c <sub>2i</sub>		—	8.0	—	pF
	Osc.4	c <sub>2i</sub>		—	8.0	—	pF
Output resistance	Osc.1	R <sub>2</sub>		—	3800	—	kΩ
	Osc.2	R <sub>2</sub>		—	65	—	kΩ
	Osc.3	R <sub>2</sub>		—	18	—	kΩ
	Osc.4	R <sub>2</sub>		—	5.0	—	kΩ

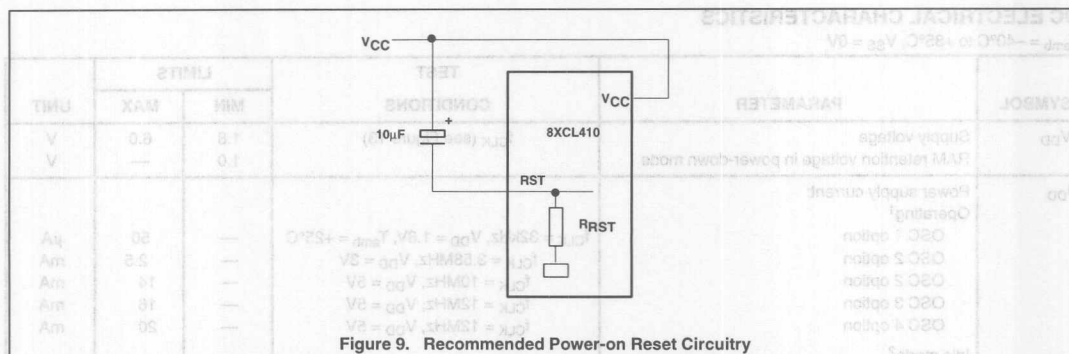
Low voltage/low power single-chip  
8-bit microcontroller with I<sup>2</sup>C

80CL410/83CL410



# Low voltage/low power single-chip 8-bit microcontroller with I<sup>2</sup>C

80CL410/83CL410



## ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>

PARAMETER	RATING	UNIT
Supply voltage	-0.5 to +6.5	V
All input voltages	-0.5 to V <sub>DD</sub> +0.5	V
DC current into any input or output	5	mA
Total power dissipation	300	mW
Storage temperature range	-65 to +150	°C
Operating ambient temperature range	-40 to +85	°C
Operating junction temperature	125	°C

### NOTES:

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
- This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
- Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V<sub>SS</sub> unless otherwise noted.

µA	±10		Input leakage current, port 0
kΩ	200	10	Internal reset pull-down resistor

### NOTES:

- The operating supply current is measured with all output pins disconnected; I<sub>ATX</sub> given with I<sub>1</sub> = I<sub>2</sub> = I<sub>3</sub> = I<sub>4</sub> = I<sub>5</sub> = I<sub>6</sub> = I<sub>7</sub> = I<sub>8</sub> = I<sub>9</sub> = I<sub>10</sub> = I<sub>11</sub> = I<sub>12</sub> = I<sub>13</sub> = I<sub>14</sub> = I<sub>15</sub> = I<sub>16</sub> = I<sub>17</sub> = I<sub>18</sub> = I<sub>19</sub> = I<sub>20</sub> = I<sub>21</sub> = I<sub>22</sub> = I<sub>23</sub> = I<sub>24</sub> = I<sub>25</sub> = I<sub>26</sub> = I<sub>27</sub> = I<sub>28</sub> = I<sub>29</sub> = I<sub>30</sub> = I<sub>31</sub> = I<sub>32</sub> = I<sub>33</sub> = I<sub>34</sub> = I<sub>35</sub> = I<sub>36</sub> = I<sub>37</sub> = I<sub>38</sub> = I<sub>39</sub> = I<sub>40</sub> = I<sub>41</sub> = I<sub>42</sub> = I<sub>43</sub> = I<sub>44</sub> = I<sub>45</sub> = I<sub>46</sub> = I<sub>47</sub> = I<sub>48</sub> = I<sub>49</sub> = I<sub>50</sub> = I<sub>51</sub> = I<sub>52</sub> = I<sub>53</sub> = I<sub>54</sub> = I<sub>55</sub> = I<sub>56</sub> = I<sub>57</sub> = I<sub>58</sub> = I<sub>59</sub> = I<sub>60</sub> = I<sub>61</sub> = I<sub>62</sub> = I<sub>63</sub> = I<sub>64</sub> = I<sub>65</sub> = I<sub>66</sub> = I<sub>67</sub> = I<sub>68</sub> = I<sub>69</sub> = I<sub>70</sub> = I<sub>71</sub> = I<sub>72</sub> = I<sub>73</sub> = I<sub>74</sub> = I<sub>75</sub> = I<sub>76</sub> = I<sub>77</sub> = I<sub>78</sub> = I<sub>79</sub> = I<sub>80</sub> = I<sub>81</sub> = I<sub>82</sub> = I<sub>83</sub> = I<sub>84</sub> = I<sub>85</sub> = I<sub>86</sub> = I<sub>87</sub> = I<sub>88</sub> = I<sub>89</sub> = I<sub>90</sub> = I<sub>91</sub> = I<sub>92</sub> = I<sub>93</sub> = I<sub>94</sub> = I<sub>95</sub> = I<sub>96</sub> = I<sub>97</sub> = I<sub>98</sub> = I<sub>99</sub> = I<sub>100</sub> = I<sub>101</sub> = I<sub>102</sub> = I<sub>103</sub> = I<sub>104</sub> = I<sub>105</sub> = I<sub>106</sub> = I<sub>107</sub> = I<sub>108</sub> = I<sub>109</sub> = I<sub>110</sub> = I<sub>111</sub> = I<sub>112</sub> = I<sub>113</sub> = I<sub>114</sub> = I<sub>115</sub> = I<sub>116</sub> = I<sub>117</sub> = I<sub>118</sub> = I<sub>119</sub> = I<sub>120</sub> = I<sub>121</sub> = I<sub>122</sub> = I<sub>123</sub> = I<sub>124</sub> = I<sub>125</sub> = I<sub>126</sub> = I<sub>127</sub> = I<sub>128</sub> = I<sub>129</sub> = I<sub>130</sub> = I<sub>131</sub> = I<sub>132</sub> = I<sub>133</sub> = I<sub>134</sub> = I<sub>135</sub> = I<sub>136</sub> = I<sub>137</sub> = I<sub>138</sub> = I<sub>139</sub> = I<sub>140</sub> = I<sub>141</sub> = I<sub>142</sub> = I<sub>143</sub> = I<sub>144</sub> = I<sub>145</sub> = I<sub>146</sub> = I<sub>147</sub> = I<sub>148</sub> = I<sub>149</sub> = I<sub>150</sub> = I<sub>151</sub> = I<sub>152</sub> = I<sub>153</sub> = I<sub>154</sub> = I<sub>155</sub> = I<sub>156</sub> = I<sub>157</sub> = I<sub>158</sub> = I<sub>159</sub> = I<sub>160</sub> = I<sub>161</sub> = I<sub>162</sub> = I<sub>163</sub> = I<sub>164</sub> = I<sub>165</sub> = I<sub>166</sub> = I<sub>167</sub> = I<sub>168</sub> = I<sub>169</sub> = I<sub>170</sub> = I<sub>171</sub> = I<sub>172</sub> = I<sub>173</sub> = I<sub>174</sub> = I<sub>175</sub> = I<sub>176</sub> = I<sub>177</sub> = I<sub>178</sub> = I<sub>179</sub> = I<sub>180</sub> = I<sub>181</sub> = I<sub>182</sub> = I<sub>183</sub> = I<sub>184</sub> = I<sub>185</sub> = I<sub>186</sub> = I<sub>187</sub> = I<sub>188</sub> = I<sub>189</sub> = I<sub>190</sub> = I<sub>191</sub> = I<sub>192</sub> = I<sub>193</sub> = I<sub>194</sub> = I<sub>195</sub> = I<sub>196</sub> = I<sub>197</sub> = I<sub>198</sub> = I<sub>199</sub> = I<sub>200</sub> = I<sub>201</sub> = I<sub>202</sub> = I<sub>203</sub> = I<sub>204</sub> = I<sub>205</sub> = I<sub>206</sub> = I<sub>207</sub> = I<sub>208</sub> = I<sub>209</sub> = I<sub>210</sub> = I<sub>211</sub> = I<sub>212</sub> = I<sub>213</sub> = I<sub>214</sub> = I<sub>215</sub> = I<sub>216</sub> = I<sub>217</sub> = I<sub>218</sub> = I<sub>219</sub> = I<sub>220</sub> = I<sub>221</sub> = I<sub>222</sub> = I<sub>223</sub> = I<sub>224</sub> = I<sub>225</sub> = I<sub>226</sub> = I<sub>227</sub> = I<sub>228</sub> = I<sub>229</sub> = I<sub>230</sub> = I<sub>231</sub> = I<sub>232</sub> = I<sub>233</sub> = I<sub>234</sub> = I<sub>235</sub> = I<sub>236</sub> = I<sub>237</sub> = I<sub>238</sub> = I<sub>239</sub> = I<sub>240</sub> = I<sub>241</sub> = I<sub>242</sub> = I<sub>243</sub> = I<sub>244</sub> = I<sub>245</sub> = I<sub>246</sub> = I<sub>247</sub> = I<sub>248</sub> = I<sub>249</sub> = I<sub>250</sub> = I<sub>251</sub> = I<sub>252</sub> = I<sub>253</sub> = I<sub>254</sub> = I<sub>255</sub> = I<sub>256</sub> = I<sub>257</sub> = I<sub>258</sub> = I<sub>259</sub> = I<sub>260</sub> = I<sub>261</sub> = I<sub>262</sub> = I<sub>263</sub> = I<sub>264</sub> = I<sub>265</sub> = I<sub>266</sub> = I<sub>267</sub> = I<sub>268</sub> = I<sub>269</sub> = I<sub>270</sub> = I<sub>271</sub> = I<sub>272</sub> = I<sub>273</sub> = I<sub>274</sub> = I<sub>275</sub> = I<sub>276</sub> = I<sub>277</sub> = I<sub>278</sub> = I<sub>279</sub> = I<sub>280</sub> = I<sub>281</sub> = I<sub>282</sub> = I<sub>283</sub> = I<sub>284</sub> = I<sub>285</sub> = I<sub>286</sub> = I<sub>287</sub> = I<sub>288</sub> = I<sub>289</sub> = I<sub>290</sub> = I<sub>291</sub> = I<sub>292</sub> = I<sub>293</sub> = I<sub>294</sub> = I<sub>295</sub> = I<sub>296</sub> = I<sub>297</sub> = I<sub>298</sub> = I<sub>299</sub> = I<sub>300</sub> = I<sub>301</sub> = I<sub>302</sub> = I<sub>303</sub> = I<sub>304</sub> = I<sub>305</sub> = I<sub>306</sub> = I<sub>307</sub> = I<sub>308</sub> = I<sub>309</sub> = I<sub>310</sub> = I<sub>311</sub> = I<sub>312</sub> = I<sub>313</sub> = I<sub>314</sub> = I<sub>315</sub> = I<sub>316</sub> = I<sub>317</sub> = I<sub>318</sub> = I<sub>319</sub> = I<sub>320</sub> = I<sub>321</sub> = I<sub>322</sub> = I<sub>323</sub> = I<sub>324</sub> = I<sub>325</sub> = I<sub>326</sub> = I<sub>327</sub> = I<sub>328</sub> = I<sub>329</sub> = I<sub>330</sub> = I<sub>331</sub> = I<sub>332</sub> = I<sub>333</sub> = I<sub>334</sub> = I<sub>335</sub> = I<sub>336</sub> = I<sub>337</sub> = I<sub>338</sub> = I<sub>339</sub> = I<sub>340</sub> = I<sub>341</sub> = I<sub>342</sub> = I<sub>343</sub> = I<sub>344</sub> = I<sub>345</sub> = I<sub>346</sub> = I<sub>347</sub> = I<sub>348</sub> = I<sub>349</sub> = I<sub>350</sub> = I<sub>351</sub> = I<sub>352</sub> = I<sub>353</sub> = I<sub>354</sub> = I<sub>355</sub> = I<sub>356</sub> = I<sub>357</sub> = I<sub>358</sub> = I<sub>359</sub> = I<sub>360</sub> = I<sub>361</sub> = I<sub>362</sub> = I<sub>363</sub> = I<sub>364</sub> = I<sub>365</sub> = I<sub>366</sub> = I<sub>367</sub> = I<sub>368</sub> = I<sub>369</sub> = I<sub>370</sub> = I<sub>371</sub> = I<sub>372</sub> = I<sub>373</sub> = I<sub>374</sub> = I<sub>375</sub> = I<sub>376</sub> = I<sub>377</sub> = I<sub>378</sub> = I<sub>379</sub> = I<sub>380</sub> = I<sub>381</sub> = I<sub>382</sub> = I<sub>383</sub> = I<sub>384</sub> = I<sub>385</sub> = I<sub>386</sub> = I<sub>387</sub> = I<sub>388</sub> = I<sub>389</sub> = I<sub>390</sub> = I<sub>391</sub> = I<sub>392</sub> = I<sub>393</sub> = I<sub>394</sub> = I<sub>395</sub> = I<sub>396</sub> = I<sub>397</sub> = I<sub>398</sub> = I<sub>399</sub> = I<sub>400</sub> = I<sub>401</sub> = I<sub>402</sub> = I<sub>403</sub> = I<sub>404</sub> = I<sub>405</sub> = I<sub>406</sub> = I<sub>407</sub> = I<sub>408</sub> = I<sub>409</sub> = I<sub>410</sub> = I<sub>411</sub> = I<sub>412</sub> = I<sub>413</sub> = I<sub>414</sub> = I<sub>415</sub> = I<sub>416</sub> = I<sub>417</sub> = I<sub>418</sub> = I<sub>419</sub> = I<sub>420</sub> = I<sub>421</sub> = I<sub>422</sub> = I<sub>423</sub> = I<sub>424</sub> = I<sub>425</sub> = I<sub>426</sub> = I<sub>427</sub> = I<sub>428</sub> = I<sub>429</sub> = I<sub>430</sub> = I<sub>431</sub> = I<sub>432</sub> = I<sub>433</sub> = I<sub>434</sub> = I<sub>435</sub> = I<sub>436</sub> = I<sub>437</sub> = I<sub>438</sub> = I<sub>439</sub> = I<sub>440</sub> = I<sub>441</sub> = I<sub>442</sub> = I<sub>443</sub> = I<sub>444</sub> = I<sub>445</sub> = I<sub>446</sub> = I<sub>447</sub> = I<sub>448</sub> = I<sub>449</sub> = I<sub>450</sub> = I<sub>451</sub> = I<sub>452</sub> = I<sub>453</sub> = I<sub>454</sub> = I<sub>455</sub> = I<sub>456</sub> = I<sub>457</sub> = I<sub>458</sub> = I<sub>459</sub> = I<sub>460</sub> = I<sub>461</sub> = I<sub>462</sub> = I<sub>463</sub> = I<sub>464</sub> = I<sub>465</sub> = I<sub>466</sub> = I<sub>467</sub> = I<sub>468</sub> = I<sub>469</sub> = I<sub>470</sub> = I<sub>471</sub> = I<sub>472</sub> = I<sub>473</sub> = I<sub>474</sub> = I<sub>475</sub> = I<sub>476</sub> = I<sub>477</sub> = I<sub>478</sub> = I<sub>479</sub> = I<sub>480</sub> = I<sub>481</sub> = I<sub>482</sub> = I<sub>483</sub> = I<sub>484</sub> = I<sub>485</sub> = I<sub>486</sub> = I<sub>487</sub> = I<sub>488</sub> = I<sub>489</sub> = I<sub>490</sub> = I<sub>491</sub> = I<sub>492</sub> = I<sub>493</sub> = I<sub>494</sub> = I<sub>495</sub> = I<sub>496</sub> = I<sub>497</sub> = I<sub>498</sub> = I<sub>499</sub> = I<sub>500</sub> = I<sub>501</sub> = I<sub>502</sub> = I<sub>503</sub> = I<sub>504</sub> = I<sub>505</sub> = I<sub>506</sub> = I<sub>507</sub> = I<sub>508</sub> = I<sub>509</sub> = I<sub>510</sub> = I<sub>511</sub> = I<sub>512</sub> = I<sub>513</sub> = I<sub>514</sub> = I<sub>515</sub> = I<sub>516</sub> = I<sub>517</sub> = I<sub>518</sub> = I<sub>519</sub> = I<sub>520</sub> = I<sub>521</sub> = I<sub>522</sub> = I<sub>523</sub> = I<sub>524</sub> = I<sub>525</sub> = I<sub>526</sub> = I<sub>527</sub> = I<sub>528</sub> = I<sub>529</sub> = I<sub>530</sub> = I<sub>531</sub> = I<sub>532</sub> = I<sub>533</sub> = I<sub>534</sub> = I<sub>535</sub> = I<sub>536</sub> = I<sub>537</sub> = I<sub>538</sub> = I<sub>539</sub> = I<sub>540</sub> = I<sub>541</sub> = I<sub>542</sub> = I<sub>543</sub> = I<sub>544</sub> = I<sub>545</sub> = I<sub>546</sub> = I<sub>547</sub> = I<sub>548</sub> = I<sub>549</sub> = I<sub>550</sub> = I<sub>551</sub> = I<sub>552</sub> = I<sub>553</sub> = I<sub>554</sub> = I<sub>555</sub> = I<sub>556</sub> = I<sub>557</sub> = I<sub>558</sub> = I<sub>559</sub> = I<sub>560</sub> = I<sub>561</sub> = I<sub>562</sub> = I<sub>563</sub> = I<sub>564</sub> = I<sub>565</sub> = I<sub>566</sub> = I<sub>567</sub> = I<sub>568</sub> = I<sub>569</sub> = I<sub>570</sub> = I<sub>571</sub> = I<sub>572</sub> = I<sub>573</sub> = I<sub>574</sub> = I<sub>575</sub> = I<sub>576</sub> = I<sub>577</sub> = I<sub>578</sub> = I<sub>579</sub> = I<sub>580</sub> = I<sub>581</sub> = I<sub>582</sub> = I<sub>583</sub> = I<sub>584</sub> = I<sub>585</sub> = I<sub>586</sub> = I<sub>587</sub> = I<sub>588</sub> = I<sub>589</sub> = I<sub>590</sub> = I<sub>591</sub> = I<sub>592</sub> = I<sub>593</sub> = I<sub>594</sub> = I<sub>595</sub> = I<sub>596</sub> = I<sub>597</sub> = I<sub>598</sub> = I<sub>599</sub> = I<sub>600</sub> = I<sub>601</sub> = I<sub>602</sub> = I<sub>603</sub> = I<sub>604</sub> = I<sub>605</sub> = I<sub>606</sub> = I<sub>607</sub> = I<sub>608</sub> = I<sub>609</sub> = I<sub>610</sub> = I<sub>611</sub> = I<sub>612</sub> = I<sub>613</sub> = I<sub>614</sub> = I<sub>615</sub> = I<sub>616</sub> = I<sub>617</sub> = I<sub>618</sub> = I<sub>619</sub> = I<sub>620</sub> = I<sub>621</sub> = I<sub>622</sub> = I<sub>623</sub> = I<sub>624</sub> = I<sub>625</sub> = I<sub>626</sub> = I<sub>627</sub> = I<sub>628</sub> = I<sub>629</sub> = I<sub>630</sub> = I<sub>631</sub> = I<sub>632</sub> = I<sub>633</sub> = I<sub>634</sub> = I<sub>635</sub> = I<sub>636</sub> = I<sub>637</sub> = I<sub>638</sub> = I<sub>639</sub> = I<sub>640</sub> = I<sub>641</sub> = I<sub>642</sub> = I<sub>643</sub> = I<sub>644</sub> = I<sub>645</sub> = I<sub>646</sub> = I<sub>647</sub> = I<sub>648</sub> = I<sub>649</sub> = I<sub>650</sub> = I<sub>651</sub> = I<sub>652</sub> = I<sub>653</sub> = I<sub>654</sub> = I<sub>655</sub> = I<sub>656</sub> = I<sub>657</sub> = I<sub>658</sub> = I<sub>659</sub> = I<sub>660</sub> = I<sub>661</sub> = I<sub>662</sub> = I<sub>663</sub> = I<sub>664</sub> = I<sub>665</sub> = I<sub>666</sub> = I<sub>667</sub> = I<sub>668</sub> = I<sub>669</sub> = I<sub>670</sub> = I<sub>671</sub> = I<sub>672</sub> = I<sub>673</sub> = I<sub>674</sub> = I<sub>675</sub> = I<sub>676</sub> = I<sub>677</sub> = I<sub>678</sub> = I<sub>679</sub> = I<sub>680</sub> = I<sub>681</sub> = I<sub>682</sub> = I<sub>683</sub> = I<sub>684</sub> = I<sub>685</sub> = I<sub>686</sub> = I<sub>687</sub> = I<sub>688</sub> = I<sub>689</sub> = I<sub>690</sub> = I<sub>691</sub> = I<sub>692</sub> = I<sub>693</sub> = I<sub>694</sub> = I<sub>695</sub> = I<sub>696</sub> = I<sub>697</sub> = I<sub>698</sub> = I<sub>699</sub> = I<sub>700</sub> = I<sub>701</sub> = I<sub>702</sub> = I<sub>703</sub> = I<sub>704</sub> = I<sub>705</sub> = I<sub>706</sub> = I<sub>707</sub> = I<sub>708</sub> = I<sub>709</sub> = I<sub>710</sub> = I<sub>711</sub> = I<sub>712</sub> = I<sub>713</sub> = I<sub>714</sub> = I<sub>715</sub> = I<sub>716</sub> = I<sub>717</sub> = I<sub>718</sub> = I<sub>719</sub> = I<sub>720</sub> = I<sub>721</sub> = I<sub>722</sub> = I<sub>723</sub> = I<sub>724</sub> = I<sub>725</sub> = I<sub>726</sub> = I<sub>727</sub> = I<sub>728</sub> = I<sub>729</sub> = I<sub>730</sub> = I<sub>731</sub> = I<sub>732</sub> = I<sub>733</sub> = I<sub>734</sub> = I<sub>735</sub> = I<sub>736</sub> = I<sub>737</sub> = I<sub>738</sub> = I<sub>739</sub> = I<sub>740</sub> = I<sub>741</sub> = I<sub>742</sub> = I<sub>743</sub> = I<sub>744</sub> = I<sub>745</sub> = I<sub>746</sub> = I<sub>747</sub> = I<sub>748</sub> = I<sub>749</sub> = I<sub>750</sub> = I<sub>751</sub> = I<sub>752</sub> = I<sub>753</sub> = I<sub>754</sub> = I<sub>755</sub> = I<sub>756</sub> = I<sub>757</sub> = I<sub>758</sub> = I<sub>759</sub> = I<sub>760</sub> = I<sub>761</sub> = I<sub>762</sub> = I<sub>763</sub> = I<sub>764</sub> = I<sub>765</sub> = I<sub>766</sub> = I<sub>767</sub> = I<sub>768</sub> = I<sub>769</sub> = I<sub>770</sub> = I<sub>771</sub> = I<sub>772</sub> = I<sub>773</sub> = I<sub>774</sub> = I<sub>775</sub> = I<sub>776</sub> = I<sub>777</sub> = I<sub>778</sub> = I<sub>779</sub> = I<sub>780</sub> = I<sub>781</sub> = I<sub>782</sub> = I<sub>783</sub> = I<sub>784</sub> = I<sub>785</sub> = I<sub>786</sub> = I<sub>787</sub> = I<sub>788</sub> = I<sub>789</sub> = I<sub>790</sub> = I<sub>791</sub> = I<sub>792</sub> = I<sub>793</sub> = I<sub>794</sub> = I<sub>795</sub> = I<sub>796</sub> = I<sub>797</sub> = I<sub>798</sub> = I<sub>799</sub> = I<sub>800</sub> = I<sub>801</sub> = I<sub>802</sub> = I<sub>803</sub> = I<sub>804</sub> = I<sub>805</sub> = I<sub>806</sub> = I<sub>807</sub> = I<sub>808</sub> = I<sub>809</sub> = I<sub>810</sub> = I<sub>811</sub> = I<sub>812</sub> = I<sub>813</sub> = I<sub>814</sub> = I<sub>815</sub> = I<sub>816</sub> = I<sub>817</sub> = I<sub>818</sub> = I<sub>819</sub> = I<sub>820</sub> = I<sub>821</sub> = I<sub>822</sub> = I<sub>823</sub> = I<sub>824</sub> = I<sub>825</sub> = I<sub>826</sub> = I<sub>827</sub> = I<sub>828</sub> = I<sub>829</sub> = I<sub>830</sub> = I<sub>831</sub> = I<sub>832</sub> = I<sub>833</sub> = I<sub>834</sub> = I<sub>835</sub> = I<sub>836</sub> = I<sub>837</sub> = I<sub>838</sub> = I<sub>839</sub> = I<sub>840</sub> = I<sub>841</sub> = I<sub>842</sub> = I<sub>843</sub> = I<sub>844</sub> = I<sub>845</sub> = I<sub>846</sub> = I<sub>847</sub> = I<sub>848</sub> = I<sub>849</sub> = I<sub>850</sub> = I<sub>851</sub> = I<sub>852</sub> = I<sub>853</sub> = I<sub>854</sub> = I<sub>855</sub> = I<sub>856</sub> = I<sub>857</sub> = I<sub>858</sub> = I<sub>859</sub> = I<sub>860</sub> = I<sub>861</sub> = I<sub>862</sub> = I<sub>863</sub> = I<sub>864</sub> = I<sub>865</sub> = I<sub>866</sub> = I<sub>867</sub> = I<sub>868</sub> = I<sub>869</sub> = I<sub>870</sub> = I<sub>871</sub> = I<sub>872</sub> = I<sub>873</sub> = I<sub>874</sub> = I<sub>875</sub> = I<sub>876</sub> = I<sub>877</sub> = I<sub>878</sub> = I<sub>879</sub> = I<sub>880</sub> = I<sub>881</sub> = I<sub>882</sub> = I<sub>883</sub> = I<sub>884</sub> = I<sub>885</sub> = I<sub>886</sub> = I<sub>887</sub> = I<sub>888</sub> = I<sub>889</sub> = I<sub>890</sub> = I<sub>891</sub> = I<sub>892</sub> = I<sub>893</sub> = I<sub>894</sub> = I<sub>895</sub> = I<sub>896</sub> = I<sub>897</sub> = I<sub>898</sub> = I<sub>899</sub> = I<sub>900</sub> = I<sub>901</sub> = I<sub>902</sub> = I<sub>903</sub> = I<sub>904</sub> = I<sub>905</sub> = I<sub>906</sub> = I<sub>907</sub> = I<sub>908</sub> = I<sub>909</sub> = I<sub>910</sub> = I<sub>911</sub> = I<sub>912</sub> = I<sub>913</sub> = I<sub>914</sub> = I<sub>915</sub> = I<sub>916</sub> = I<sub>917</sub> = I<sub>918</sub> = I<sub>919</sub> = I<sub>920</sub> = I<sub>921</sub> = I<sub>922</sub> = I<sub>923</sub> = I<sub>924</sub> = I<sub>925</sub> = I<sub>926</sub> = I<sub>927</sub> = I<sub>928</sub> = I<sub>929</sub> = I<sub>930</sub> = I<sub>931</sub> = I<sub>932</sub> = I<sub>933</sub> = I<sub>934</sub> = I<sub>935</sub> = I<sub>936</sub> = I<sub>937</sub> = I<sub>938</sub> = I<sub>939</sub> = I<sub>940</sub> = I<sub>941</sub> = I<sub>942</sub> = I<sub>943</sub> = I<sub>944</sub> = I<sub>945</sub> = I<sub>946</sub> = I<sub>947</sub> = I<sub>948</sub> = I<sub>949</sub> = I<sub>950</sub> = I<sub>951</sub> = I<sub>952</sub> = I<sub>953</sub> = I<sub>954</sub> = I<sub>955</sub> = I<sub>956</sub> = I<sub>957</sub> = I<sub>958</sub> = I<sub>959</sub> = I<sub>960</sub> = I<sub>961</sub> = I<sub>962</sub> = I<sub>963</sub> = I<sub>964</sub> = I<sub>965</sub> = I<sub>966</sub> = I<sub>967</sub> = I<sub>968</sub> = I<sub>969</sub> = I<sub>970</sub> = I<sub>971</sub> = I<sub>972</sub> = I<sub>973</sub> = I<sub>974</sub> = I<sub>975</sub> = I<sub>976</sub> = I<sub>977</sub> = I<sub>978</sub> = I<sub>979</sub> = I<sub>980</sub> = I<sub>981</sub> = I<sub>982</sub> = I<sub>983</sub> = I<sub>984</sub> = I<sub>985</sub> = I<sub>986</sub> = I<sub>987</sub> = I<sub>988</sub> = I<sub>989</sub> = I<sub>990</sub> = I<sub>991</sub> = I<sub>992</sub> = I<sub>993</sub> = I<sub>994</sub> = I<sub>995</sub> = I<sub>996</sub> = I<sub>997</sub> = I<sub>998</sub> = I<sub>999</sub> = I<sub>1000</sub> = I<sub>1001</sub> = I<sub>1002</sub> = I<sub>1003</sub> = I<sub>1004</sub> = I<sub>1005</sub> = I<sub>1006</sub> = I<sub>1007</sub> = I<sub>1008</sub> = I<sub>1009</sub> = I<sub>1010</sub> = I<sub>1011</sub> = I<sub>1012</sub> = I<sub>1013</sub> = I<sub>1014</sub> = I<sub>1015</sub> = I<sub>1016</sub> = I<sub>1017</sub> = I<sub>1018</sub> = I<sub>1019</sub> = I<sub>1020</sub> = I<sub>1021</sub> = I<sub>1022</sub> = I<sub>1023</sub> = I<sub>1024</sub> = I<sub>1025</sub> = I<sub>1026</sub> = I

## Low voltage/low power single-chip 8-bit microcontroller with I<sup>2</sup>C

80CL410/83CL410

## DC ELECTRICAL CHARACTERISTICS

 $T_{amb} = -40^{\circ}\text{C to } +85^{\circ}\text{C}, V_{SS} = 0\text{V}$ 

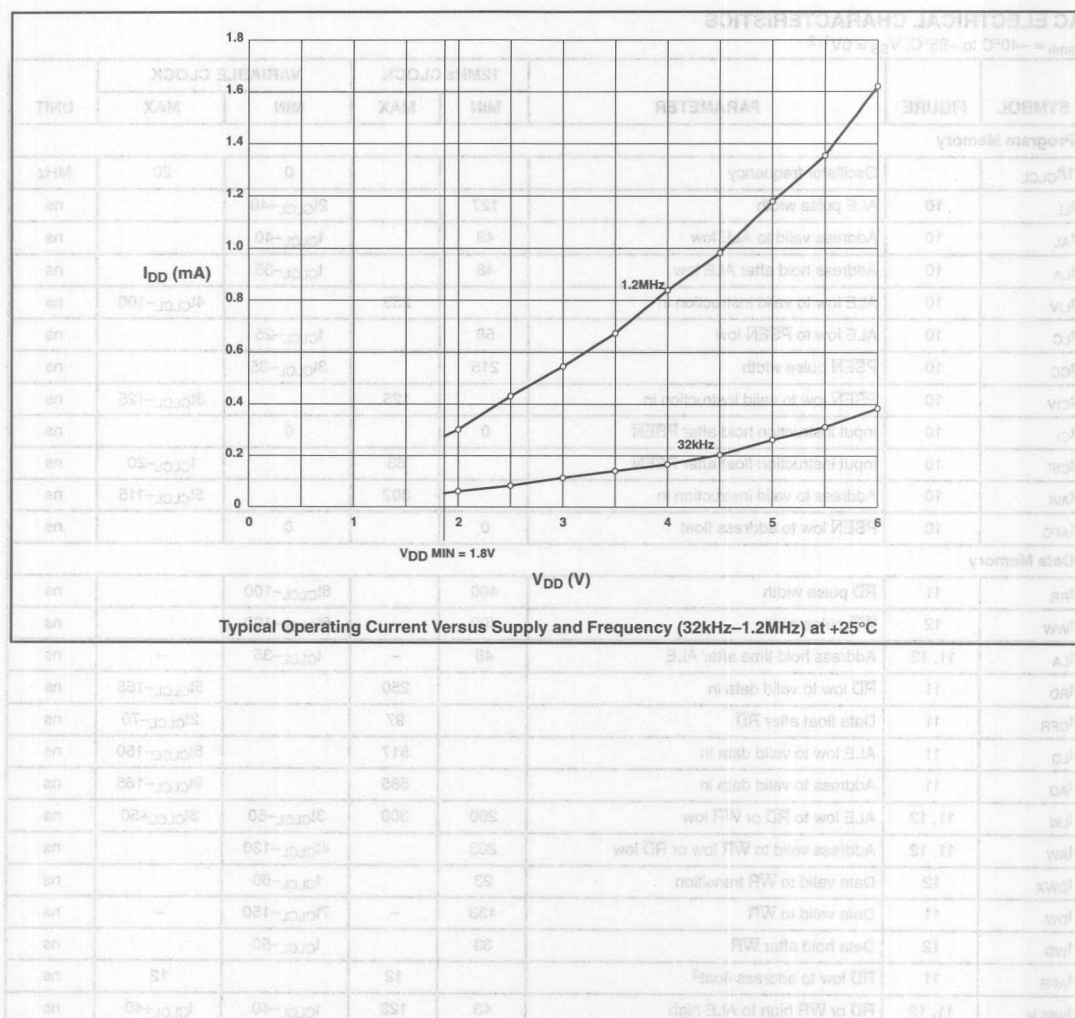
SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			MIN	MAX	
V <sub>DD</sub>	Supply voltage RAM retention voltage in power-down mode	f <sub>CLK</sub> (see Figure 13)	1.8 1.0	6.0 —	V V
I <sub>DD</sub>	Power supply current: Operating <sup>1</sup> OSC 1 option OSC 2 option OSC 2 option OSC 3 option OSC 4 option  Idle mode <sup>2</sup> OSC 1 option OSC 2 option OSC 2 option OSC 3 option OSC 4 option  Power-down mode <sup>3</sup>	f <sub>CLK</sub> = 32kHz, V <sub>DD</sub> = 1.8V, T <sub>amb</sub> = +25°C f <sub>CLK</sub> = 3.58MHz, V <sub>DD</sub> = 3V f <sub>CLK</sub> = 10MHz, V <sub>DD</sub> = 5V f <sub>CLK</sub> = 12MHz, V <sub>DD</sub> = 5V f <sub>CLK</sub> = 12MHz, V <sub>DD</sub> = 5V  f <sub>CLK</sub> = 32kHz, V <sub>DD</sub> = 1.8V, T <sub>amb</sub> = +25°C f <sub>CLK</sub> = 3.58MHz, V <sub>DD</sub> = 3V f <sub>CLK</sub> = 10MHz, V <sub>DD</sub> = 5V f <sub>CLK</sub> = 12MHz, V <sub>DD</sub> = 5V f <sub>CLK</sub> = 12MHz, V <sub>DD</sub> = 5V  V <sub>DD</sub> = 1.8V, T <sub>amb</sub> = +25°C	— — — — —  — — — — —  —	50 2.5 14 16 20  25 1.0 5.0 7.0 8.5 10	μA mA mA mA mA  μA mA mA mA mA  μA
V <sub>IL</sub>	Input low voltage		V <sub>SS</sub>	0.3V <sub>DD</sub>	V
V <sub>IH</sub>	Input high voltage		0.7V <sub>DD</sub>	V <sub>DD</sub>	V
I <sub>OL</sub>	Output sink current, except SDA, SCL	V <sub>DD</sub> = 5V, V <sub>OL</sub> = 0.4V V <sub>DD</sub> = 2.5V, V <sub>OL</sub> = 0.4V	1.6 0.7		mA mA
I <sub>OL1</sub>	Output sink current, SDA, SCL	V <sub>DD</sub> = 5V, V <sub>OL</sub> = 0.4V	3.0		mA
I <sub>OH</sub>	Output source current (push-pull options only)	V <sub>DD</sub> = 5V, V <sub>OH</sub> = V <sub>DD</sub> - 0.4V V <sub>DD</sub> = 2.5V, V <sub>OH</sub> = V <sub>DD</sub> - 0.4V	1.6 0.7		mA mA
I <sub>IL</sub>	Logical 0 input current, ports 1, 2, 3	V <sub>DD</sub> = 5V, V <sub>IN</sub> = 0.4V V <sub>DD</sub> = 2.5V, V <sub>IN</sub> = 0.4V		-100 -50	μA μA
I <sub>TL</sub>	Logical 1-to-0 transition current, ports 1, 2, 3	V <sub>DD</sub> = 5V, V <sub>IN</sub> = V <sub>DD</sub> /2 V <sub>DD</sub> = 2.5V, V <sub>IN</sub> = V <sub>DD</sub> /2		-1.0 -500	mA μA
I <sub>LI</sub>	Input leakage current, port 0, E <sub>A</sub>	V <sub>SS</sub> < V <sub>I</sub> < V <sub>DD</sub>		±10	μA
R <sub>RST</sub>	Internal reset pull-down resistor		10	200	kΩ

NOTES:

1. The operating supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 10\text{ns}$ ;  $V_{IL} = V_{SS}$ ,  $V_{IH} = V_{DD}$ ; XTAL2 not connected;  $\overline{EA} = RST = \text{Port } 0 = V_{DD}$ ; all open drain outputs connected to  $V_{SS}$ .
2. The idle supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 10\text{ns}$ ;  $V_{IL} = V_{SS}$ ,  $V_{IH} = V_{DD}$ ; XTAL2 not connected;  $\overline{EA} = \text{Port } 0 = V_{DD}$ ;  $RST = V_{SS}$ ; all open drain outputs connected to  $V_{SS}$ .
3. The power-down current is measured with all output pins disconnected; XTAL1 not connected;  $\overline{EA} = \text{port } 0 = V_{DD}$ ;  $RST = V_{SS}$ ; all open-drain outputs connected to  $V_{SS}$ .
4. The RC-oscillator is not implemented in this version.
5. Circuits with "power-on reset" option "OFF" are tested at  $V_{DDMIN} = 1.8\text{V}$ , with option "ON" (typically 1.3V) are tested at  $V_{DDMIN} = 2.3\text{V}$ .

# Low voltage/low power single-chip 8-bit microcontroller with I<sup>2</sup>C

80CL410/83CL410



NOTES:

1. Parameters are valid over operating temperature range unless otherwise specified.
2. Load capacitance for port 0, ALE, and PSEN = 50pF, load capacitance for all other outputs = 40pF.
3. Interfacing the 80CL410 to devices with four time up to 70ns is permitted. This limited bus connection will not cause damage to port 0 driver.



# Low voltage/low power single-chip 8-bit microcontroller with I<sup>2</sup>C

80CL410/83CL410

## AC ELECTRICAL CHARACTERISTICS

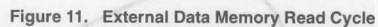
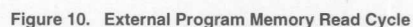
T<sub>amb</sub> = -40°C to +85°C, V<sub>SS</sub> = 0V<sup>1,2</sup>

SYMBOL	FIGURE	PARAMETER	12MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
Program Memory							
1/t <sub>CLCL</sub>		Oscillator frequency			0	20	MHz
t <sub>LL</sub>	10	ALE pulse width	127		2t <sub>CLCL</sub> -40		ns
t <sub>AL</sub>	10	Address valid to ALE low	43		t <sub>CLCL</sub> -40		ns
t <sub>LA</sub>	10	Address hold after ALE low	48		t <sub>CLCL</sub> -35		ns
t <sub>LIV</sub>	10	ALE low to valid instruction in		233		4t <sub>CLCL</sub> -100	ns
t <sub>LC</sub>	10	ALE low to PSEN low	58		t <sub>CLCL</sub> -25		ns
t <sub>CC</sub>	10	PSEN pulse width	215		3t <sub>CLCL</sub> -35		ns
t <sub>CIV</sub>	10	PSEN low to valid instruction in		125		3t <sub>CLCL</sub> -125	ns
t <sub>CI</sub>	10	Input instruction hold after PSEN	0		0		ns
t <sub>CIF</sub>	10	Input instruction float after PSEN		63		t <sub>CLCL</sub> -20	ns
t <sub>AVI</sub>	10	Address to valid instruction in		302		5t <sub>CLCL</sub> -115	ns
t <sub>AFC</sub>	10	PSEN low to address float	0		0		ns
Data Memory							
t <sub>RR</sub>	11	RD pulse width	400		6t <sub>CLCL</sub> -100		ns
t <sub>WW</sub>	12	WR pulse width	400		6t <sub>CLCL</sub> -100		ns
t <sub>LA</sub>	11, 12	Address hold time after ALE	48	-	t <sub>CLCL</sub> -35	-	ns
t <sub>RD</sub>	11	RD low to valid data in		250		5t <sub>CLCL</sub> -165	ns
t <sub>DFR</sub>	11	Data float after RD		97		2t <sub>CLCL</sub> -70	ns
t <sub>LD</sub>	11	ALE low to valid data in		517		8t <sub>CLCL</sub> -150	ns
t <sub>AD</sub>	11	Address to valid data in		585		9t <sub>CLCL</sub> -165	ns
t <sub>LW</sub>	11, 12	ALE low to RD or WR low	200	300	3t <sub>CLCL</sub> -50	3t <sub>CLCL</sub> +50	ns
t <sub>AW</sub>	11, 12	Address valid to WR low or RD low	203		4t <sub>CLCL</sub> -130		ns
t <sub>DWX</sub>	12	Data valid to WR transition	23		t <sub>CLCL</sub> -60		ns
t <sub>DW</sub>	11	Data valid to WR	433	-	7t <sub>CLCL</sub> -150	-	ns
t <sub>WD</sub>	12	Data hold after WR	33		t <sub>CLCL</sub> -50		ns
t <sub>AFR</sub>	11	RD low to address float <sup>3</sup>		12		12	ns
t <sub>WHLH</sub>	11, 12	RD or WR high to ALE high	43	123	t <sub>CLCL</sub> -40	t <sub>CLCL</sub> +40	ns

### NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 50pF, load capacitance for all other outputs = 40pF.
- Interfacing the 8XCL410 to devices with float time up to 75ns is permitted. This limited bus connection will not cause damage to port 0 drivers.

## 80CL410/83CL410



# Low voltage/low power single-chip 8-bit microcontroller with I<sup>2</sup>C

80CL410/83CL410

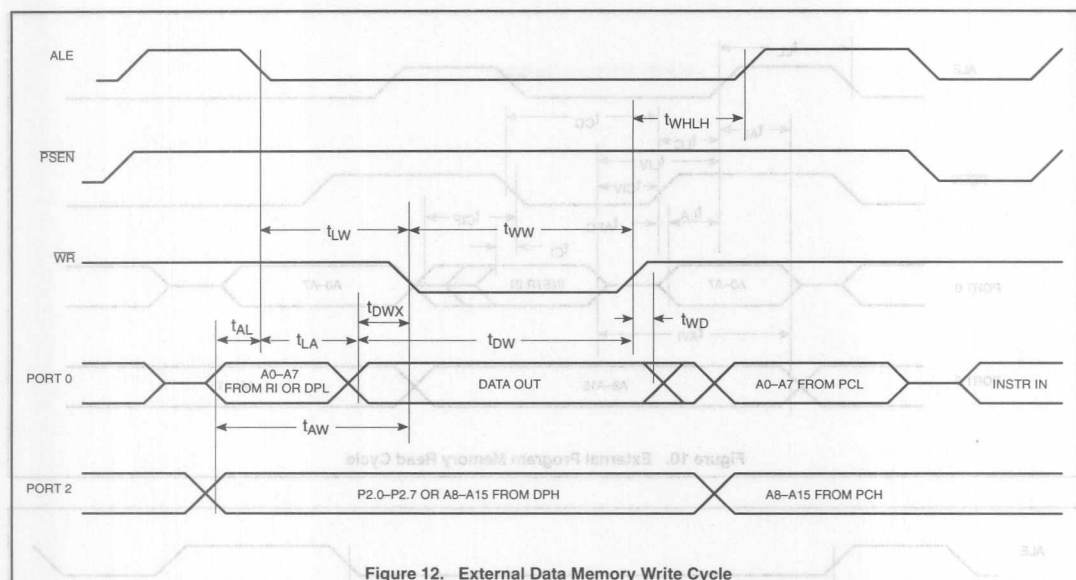


Figure 12. External Data Memory Write Cycle

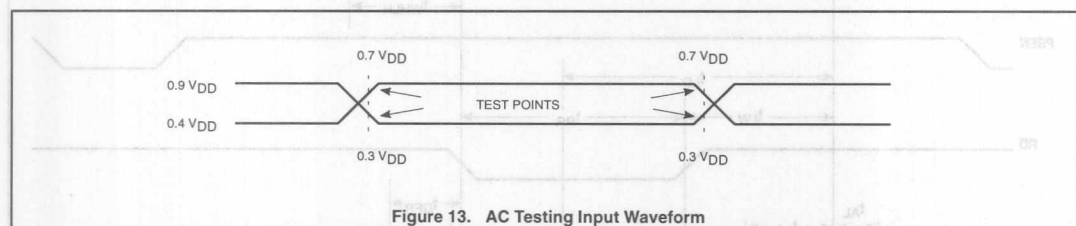


Figure 13. AC Testing Input Waveform

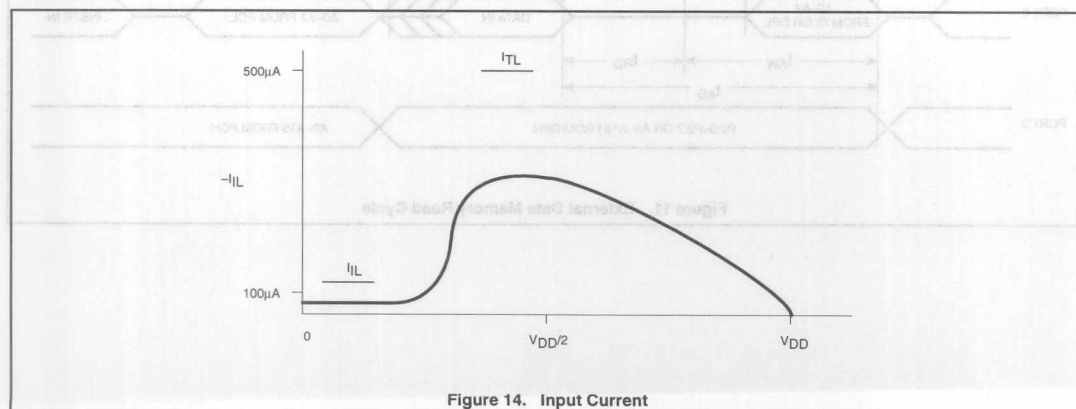
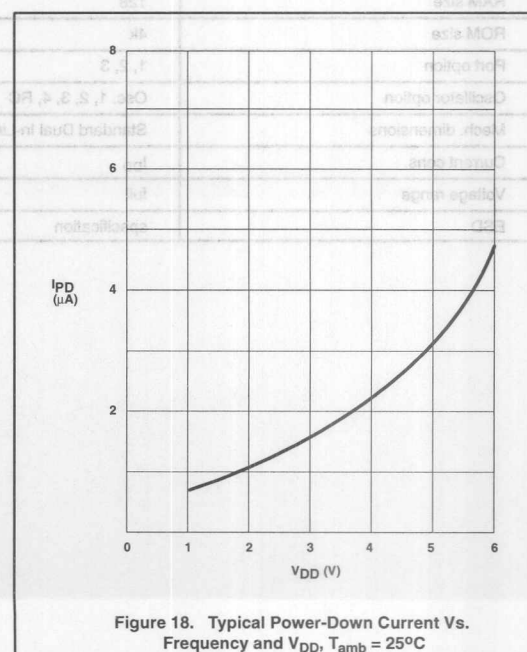
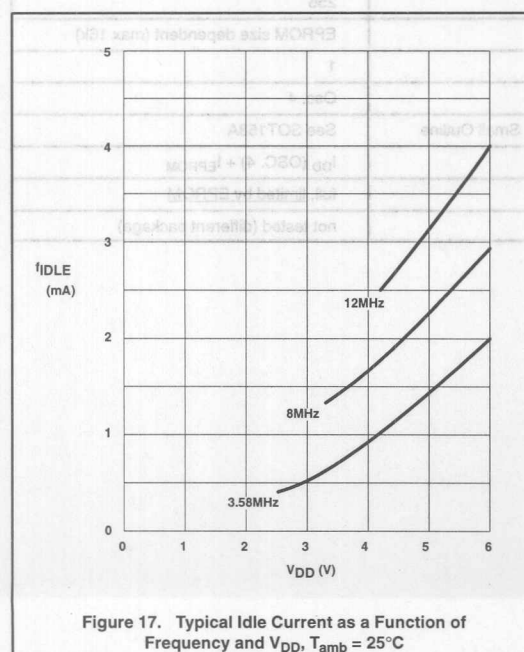
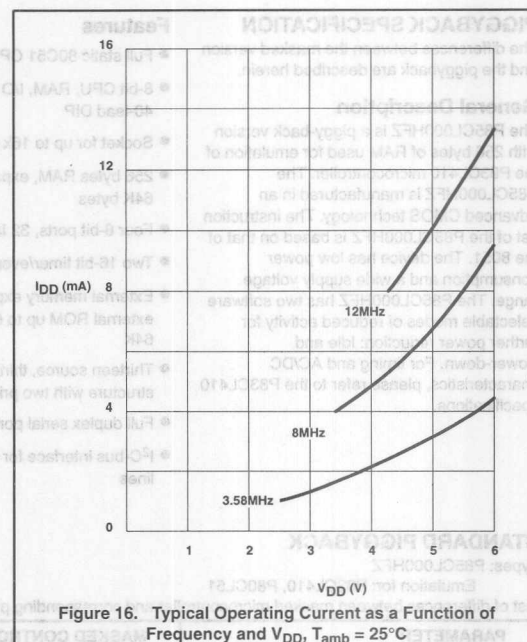
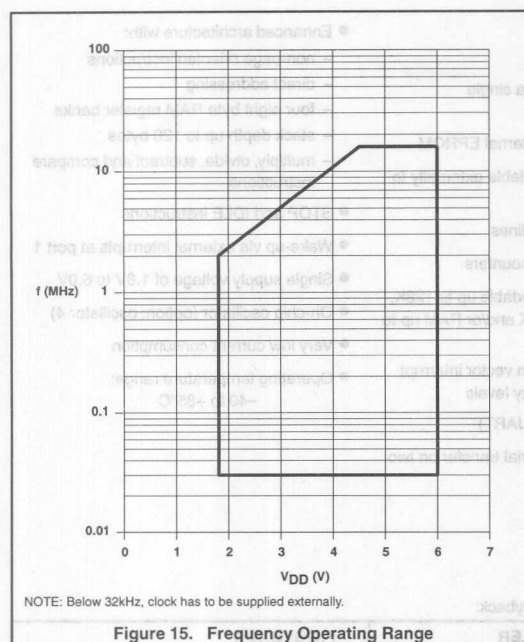


Figure 14. Input Current

# Low voltage/low power single-chip 8-bit microcontroller with I<sup>2</sup>C

80CL410/83CL410



## Low voltage/low power single-chip 8-bit microcontroller with I<sup>2</sup>C

# 80CL410/83CL410

### PIGGYBACK SPECIFICATION

The differences between the masked version and the piggyback are described herein.

### General Description

The P85CL000HFZ is a piggy-back version with 256 bytes of RAM used for emulation of the P83CL410 microcontroller. The P85CL000HFZ is manufactured in an advanced CMOS technology. The instruction set of the P85CL000HFZ is based on that of the 8051. The device has low power consumption and a wide supply voltage range. The P85CL000HFZ has two software selectable modes of reduced activity for further power reduction: Idle and Power-down. For timing and AC/DC characteristics, please refer to the P83CL410 specifications.

### Features

- Full static 80C51 CPU
- 8-bit CPU, RAM, I/O in a single 40-lead DIP
- Socket for up to 16k external EPROM
- 256 bytes RAM, expandable externally to 64K bytes
- Four 8-bit ports, 32 I/O lines
- Two 16-bit timer/event counters
- External memory expandable up to 128K, external ROM up to 64K and/or RAM up to 64K
- Thirteen source, thirteen vector interrupt structure with two priority levels
- Full duplex serial port (UART)
- I<sup>2</sup>C-bus interface for serial transfer on two lines

### Enhanced architecture with:

- non-page oriented instructions
- direct addressing
- four eight byte RAM register banks
- stack depth up to 128 bytes
- multiply, divide, subtract and compare instructions
- STOP and IDLE instructions
- Wake-up via external interrupts at port 1
- Single supply voltage of 1.8V to 6.0V
- On-chip oscillator (option: oscillator 4)
- Very low current consumption
- Operating temperature range: -40 to +85°C

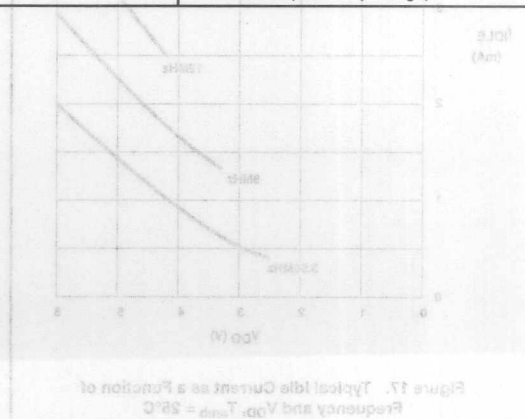
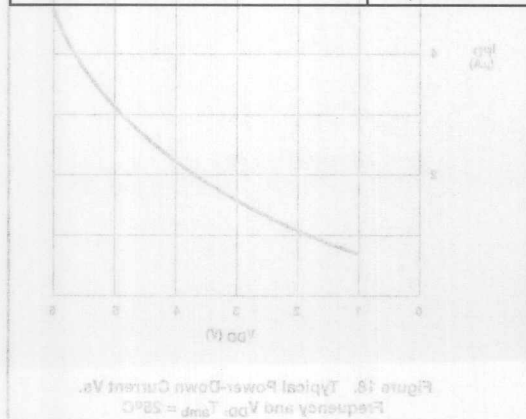
### STANDARD PIGGYBACK

Types: P85CL000HFZ

Emulation for: P83CL410, P80CL51

List of differences between masked microcontroller and corresponding piggyback:

PARAMETER	MASKED CONTROLLER	PIGGYBACK
RAM size	128	256
ROM size	4k	EPROM size dependent (max 16k)
Port option	1, 2, 3	1
Oscillator option	Osc. 1, 2, 3, 4, RC	Osc. 4
Mech. dimensions	Standard Dual In-Line, Small Outline	See SOT158A
Current cons.	I <sub>DD</sub>	I <sub>DD</sub> (OSC. 4) + I <sub>EPROM</sub>
Voltage range	full	full, limited by EPROM
ESD	specification	not tested (different package)





# CMOS single-chip 8-bit microcontroller 80C451/83C451/87C451

## DESCRIPTION

The Philips 8XC451 is an I/O expanded single-chip microcontroller fabricated with Philips high-density CMOS technology. Philips epitaxial substrate minimizes latch-up sensitivity.

The 8XC451 (includes the 80C451, 87C451 and 83C451) is a functional extension of the 87C51 microcontroller with three additional I/O ports and four I/O control lines for a total of 68 pins. Four control lines associated with port 6 facilitate high-speed asynchronous I/O functions.

The 8XC451 includes a 4k × 8 ROM (83C451) EPROM (87C451), a 128 × 8 RAM, 56 (LCC), two 16-bit timer/counters, a five source, two priority level, nested interrupt structure, a serial I/O port for either a full duplex UART, I/O expansion, or multi-processor communications, and on-chip oscillator and clock circuits. The 80C451 ROMless version includes all of the 83C451 features except the on-board 4k × 8 ROM.

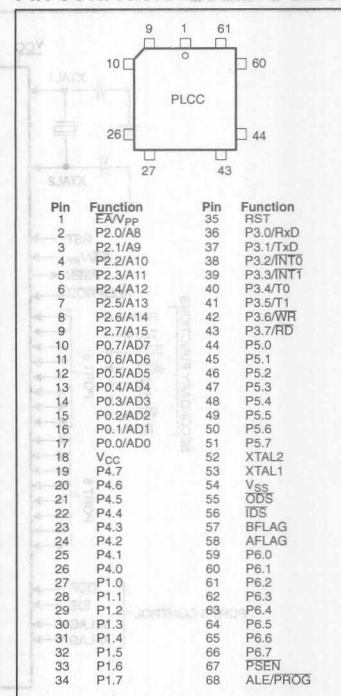
The 87C451 has 4k of EPROM on-chip as program memory and is otherwise identical to the 83C451.

The 8XC451 has two software selectable modes of reduced activity for further power reduction; idle mode and power-down mode. Idle mode freezes the CPU while allowing the RAM, timers, serial port, and interrupt system to continue functioning. Power-down mode freezes the oscillator, causing all other chip functions to be inoperative while maintaining the RAM contents.

## FEATURES

- 80C51 based architecture
- Seven 8-bit I/O ports
- Port 6 features:
  - Eight data pins
  - Four control pins
  - Direct MPU bus interface
  - Parallel printer interface
- On the microcontroller:
  - 4k × 8 ROM (83C451)
  - 4k × 8 EPROM (87C451)
  - ROMless version (80C451)
  - 128 × 8 RAM
  - Two 16-bit counter/timers
  - Two external interrupts
- External memory addressing capability
  - 64k ROM and 64k RAM
- Low power consumption:
  - Normal operation: less than 24mA at 5V, 12MHz
  - Idle mode
  - Power-down mode

## PIN CONFIGURATION



## ORDERING INFORMATION

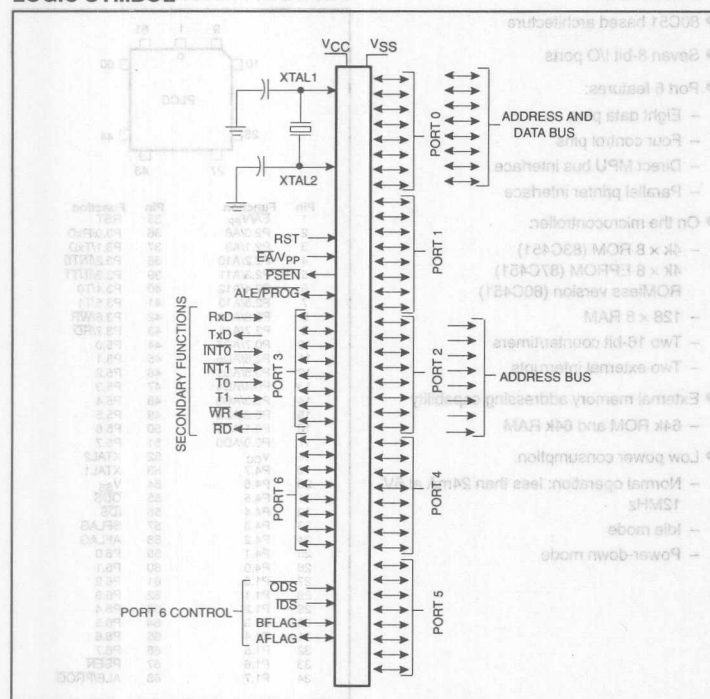
ROMless	ROM	EPROM <sup>1</sup>	TEMPERATURE RANGE °C AND PACKAGE	FREQ MHz	DRAWING NUMBER
SC80C451CCA68	SC83C451CCA68	SC87C451CCA68	OTP 0 to +70, Plastic Leaded Chip Carrier,	3.5 to 12	0398E
SC80C451CGA68	SC83C451CGA68	SC87C451CGA68	OTP 0 to +70, Plastic Leaded Chip Carrier	3.5 to 16	0398E
SC80C451ACA68	SC83C451ACA68	SC87C451ACA68	OTP -40 to +85, Plastic Leaded Chip Carrier	3.5 to 12	0398E
SC80C451AGA68	SC83C451AGA68	SC87C451AGA68	OTP -40 to +85, Plastic Leaded Chip Carrier	3.5 to 16	0398E

### NOTE:

1. OTP = One Time Programmable

## CMOS single-chip 8-bit microcontroller 80C451/83C451/87C451

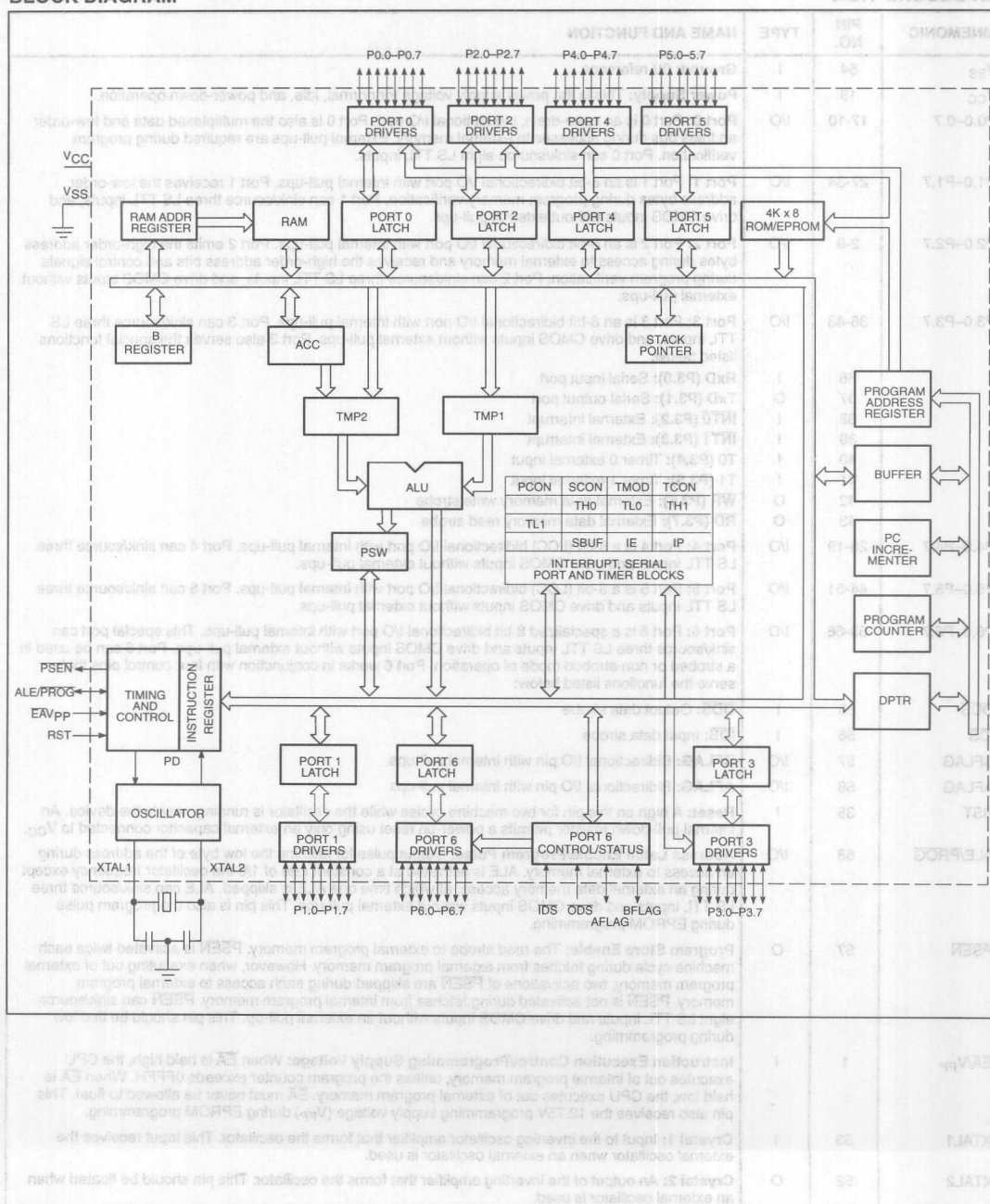
**LOGIC SYMBOL**



- Normal operation: less than 120mW
  - Idle mode
  - Power-down mode
- External memory addressing
- 512K ROM and 512K RAM
  - Two external interrupts
  - Two 16-bit counters
  - 128 x 8 RAM
- ROMless version (80C43)
- 4K x 8 PROM (87C43)
  - 8K x 8 PROM (83C43)
- On the microcontroller:
- Parallel printer interface
  - Direct MPU bus interface
  - Four control pins
  - Eight data pins
- ADDRESS AND DATA BUS**
- Port E features:
- Seven 8-bit I/O ports
  - 80C51 based architecture

## CMOS single-chip 8-bit microcontroller 80C451/83C451/87C451

## BLOCK DIAGRAM



## CMOS single-chip 8-bit microcontroller 80C451/83C451/87C451

## PIN DESCRIPTION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
V <sub>SS</sub>	54	I	<b>Ground:</b> 0V reference.
V <sub>CC</sub>	18	I	<b>Power Supply:</b> This is the power supply voltage for normal, idle, and power-down operation.
P0.0–P0.7	17-10	I/O	<b>Port 0:</b> Port 0 is an open-drain, bidirectional I/O port. Port 0 is also the multiplexed data and low-order address bus during accesses to external memory. External pull-ups are required during program verification. Port 0 can sink/source eight LS TTL inputs.
P1.0–P1.7	27-34	I/O	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. Port 1 receives the low-order address bytes during program memory verification. Port 1 can sink/source three LS TTL inputs, and drive CMOS inputs without external pull-ups.
P2.0–P2.7	2-9	I/O	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 emits the high-order address bytes during access to external memory and receives the high-order address bits and control signals during program verification. Port 2 can sink/source three LS TTL inputs, and drive CMOS inputs without external pull-ups.
P3.0–P3.7	36-43	I/O	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 can sink/source three LS TTL inputs, and drive CMOS inputs without external pull-ups. Port 3 also serves the special functions listed below:
	36	I	<b>RxD (P3.0):</b> Serial input port
	37	O	<b>TxD (P3.1):</b> Serial output port
	38	I	<b>INT0 (P3.2):</b> External interrupt
	39	I	<b>INT1 (P3.3):</b> External interrupt
	40	I	<b>T0 (P3.4):</b> Timer 0 external input
	41	I	<b>T1 (P3.5):</b> Timer 1 external input
	42	O	<b>WR (P3.6):</b> External data memory write strobe
	43	O	<b>RD (P3.7):</b> External data memory read strobe
P4.0–P4.7	26-19	I/O	<b>Port 4:</b> Port 4 is a 8-bit (LCC) bidirectional I/O port with internal pull-ups. Port 4 can sink/source three LS TTL inputs and drive CMOS inputs without external pull-ups.
P5.0–P5.7	44-51	I/O	<b>Port 5:</b> Port 5 is a 8-bit (LCC) bidirectional I/O port with internal pull-ups. Port 5 can sink/source three LS TTL inputs and drive CMOS inputs without external pull-ups.
P6.0–P6.7	59-66	I/O	<b>Port 6:</b> Port 6 is a specialized 8-bit bidirectional I/O port with internal pull-ups. This special port can sink/source three LS TTL inputs and drive CMOS inputs without external pull-ups. Port 6 can be used in a strobed or non-strobed mode of operation. Port 6 works in conjunction with four control pins that serve the functions listed below:
ODS	55	I	<b>ODS:</b> Output data strobe
IDS	56	I	<b>IDS:</b> Input data strobe
BFLAG	57	I/O	<b>BFLAG:</b> Bidirectional I/O pin with internal pull-ups
AFLAG	58	I/O	<b>AFLAG:</b> Bidirectional I/O pin with internal pull-ups
RST	35	I	<b>Reset:</b> A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal pull-down resistor permits a power-on reset using only an external capacitor connected to V <sub>CC</sub> .
ALE/PROG	68	I/O	<b>Address Latch Enable/Program Pulse:</b> Output pulse for latching the low byte of the address during an access to external memory. ALE is activated at a constant rate of 1/6 the oscillator frequency except during an external data memory access, at which time one ALE is skipped. ALE can sink/source three LS TTL inputs and drive CMOS inputs without external pull-ups. This pin is also the program pulse during EPROM programming.
PSEN	67	O	<b>Program Store Enable:</b> The read strobe to external program memory. PSEN is activated twice each machine cycle during fetches from external program memory. However, when executing out of external program memory, two activations of PSEN are skipped during each access to external program memory. PSEN is not activated during fetches from internal program memory. PSEN can sink/source eight LS TTL inputs and drive CMOS inputs without an external pull-up. This pin should be tied low during programming.
EA/V <sub>PP</sub>	1	I	<b>Instruction Execution Control/Programming Supply Voltage:</b> When EA is held high, the CPU executes out of internal program memory, unless the program counter exceeds 0FFFH. When EA is held low, the CPU executes out of external program memory. EA must never be allowed to float. This pin also receives the 12.75V programming supply voltage (V <sub>PP</sub> ) during EPROM programming.
XTAL1	53	I	<b>Crystal 1:</b> Input to the inverting oscillator amplifier that forms the oscillator. This input receives the external oscillator when an external oscillator is used.
XTAL2	52	O	<b>Crystal 2:</b> An output of the inverting amplifier that forms the oscillator. This pin should be floated when an external oscillator is used.

## 80C451/£3C451

ven parallel I/O through P3, are present on the s 4 and 5 are that is, they are ports with no standard output a specialized internal

I/O ports with 8-bit port. Port 4 is connected to them, are 8-bit ports, and in function register

ational I/O port (Figure 1). This three LS TTL without of this port data can be used as a and modes of four special BFLAG, and BFLAG controlled by the (CSR). Port 6 at the special shown in Table 1. are used in

active Low) for  
used to control  
the output buffer  
the OBF flag  
(mode).  $\overline{\text{ODS}}$  is  
control. the OBF  
cleared on the  
 $\overline{\text{DS}}$ .

**IDS** – Input data strobe (Active Low) for port 6. **IDS** is used to control the port 6 input latch and input buffer full flag (IBF) bit in the CSR. The input data latch can be programmed to be transparent when **IDS** is low and latched on the positive transition of **IDS**, or to latch only on the positive transition of **IDS**. Correspondingly, the IBF flag is set on the negative or positive transition of **IDS**.

**BFLAG** – BFLAG is a bidirectional I/O pin which can be programmed to be an output, set high or low under program control, or to output the state of the input buffer full flag. BFLAG can also be programmed to input an enable signal for port 6. When BFLAG is used as an enable input, port 6 output drivers are in the high-impedance state, and the input latch does not respond to the  $\overline{TD\overline{S}}$  strobe when BFLAG is high. Both features are enabled when BFLAG is low. This feature facilitates the use of the SC8XC451 in bused multiprocessor systems.

**AFLAG** – AFLAG is a bidirectional I/O pin which can be programmed to be an output set high or low under program control, or to output the state of the output buffer full flag. AFLAG can also be programmed to be an input which selects whether the contents of the output buffer, or the contents of the port 6 control status register will output on port 6. This feature grants complete port 6 status to external devices.

Port 6 can be used in a number of different ways to facilitate data communication. It can be used as a processor bus interface, as a standard quasi-bidirectional I/O port, or as a parallel printer port (either polled or interrupt driven).

## interface

an 8XC451 as  
processor type bus.  
a general purp  
a microcontrolle  
ing up the 8XC4

a processor bus interface allows multiple microcontrollers to be used as flexible peripheral processing elements. Applications can include: keyboard scanner, serial I/O controllers, servo controllers, etc.

On reset, port 6 is programmed correctly (that is, Special Function registers CSR and P6) for use as a bus interface. This prevents the interface from disrupting data on the bus of a host processor during power-up.

### Standard Quasi-bidirectional I/O Port

To use port 6 as a common I/O port, all of the control pins should be tied to ground. On hardware reset, bits 2-7 of the CSR are set to one. With the control pins grounded, the port's operation and electrical characteristics will be identical to port 1 on the 80C51. No further software initialization is required.

### Parallel Printer Port

The BXC451 has the capacity to permit all of the intelligent features of a common printer to be handled by a single chip. The features of port 6 allow a parallel port to be designed with only line driving and receiving chips required as additional hardware. The onboard UART allows RS232C interfacing with only level shifting chips added. The 8-bit parallel ports 0 to 6 are ample to drive onboard control functions, even when ports are used for external memory access, interrupts, and other functions. The RAM addressing ability of ports 0 to 2 can be used to address up to 64k bytes of a hardware buffer/spooler.

In addition, either end of a parallel interface can be implemented using port 6, and the interfaces can be interrupt driven or polled in either case. For more detailed information on port 6 usage, refer to the application notes entitled "80C451 Operation of Port 6" and "256k Centronics Printer Buffer Using the SC87C451 Microcontroller."



## CMOS single-chip 8-bit microcontroller 80C451/83C451/87C451

**CONTROL STATUS REGISTER**

The control status register (CSR) establishes the mode of operation for port 6 and indicates the current status of port 6 I/O registers. All control status register bits can be read and written by the CPU, except bits 0 and 1, which are read only. Reset writes ones to bits 2 through 7, and writes zeros to bits 0 and 1 (see Table 3).

**CSR.0 Input Buffer Full Flag (IBF) (Read Only)** – The IBF bit is set to a logic 1 when port 6 data is loaded into the input buffer under control of  $\overline{IDS}$ . This can occur on the negative or positive edge of  $\overline{IDS}$ , as determined by CSR.2 IBF is cleared when the CPU reads the input buffer register.

**CSR.1 Output Buffer Full Flag (OBF) (Read Only)** – The OBF flag is set to a logic 1 when the CPU writes to the port 6 output data buffer. OBF is cleared by the positive or negative edge of  $\overline{ODS}$ , as determined by CSR.3.

**CSR.2 IDS Mode Select (IDSM)** – When CSR.2 = 0, a low-to-high transition on the  $\overline{IDS}$  pin sets the IBF flag. The Port 6 input buffer is loaded on the  $\overline{IDS}$  positive edge.

When CSR.2 = 1, a high-to-low transition on the  $\overline{IDS}$  pin sets the IBF flag. Port 6 input buffer is transparent when  $\overline{IDS}$  is low, and latched when  $\overline{IDS}$  is high.

**CSR.3 Output Buffer Full Flag Clear Mode (OBFCL)** – When CSR.3 = 1, the positive edge of the  $\overline{ODS}$  input clears the OBF flag. When CSR.3 = 0, the negative edge of the  $\overline{ODS}$  input clears the OBF flag.

**CSR.4, CSR.5 AFLAG Mode Select (MA0, MA1)** – Bits 4 and 5 select the mode of operation for the AFLAG pin as follows:

MA1	MA0	AFLAG Function
0	0	Logic 0 output
0	1	Logic 1 output
1	0	OBF flag output (CSR.1)
1	1	Select (SEL) input mode

The select (SEL) input mode is used to determine whether the port 6 data register or the control status register is output on port 6. When the select feature is enabled, the AFLAG input controls the source of port 6 output data. A logic 0 on AFLAG input selects the port 6 data register, and a logic 1 on AFLAG input selects the control status register.

**CSR.6, CSR.7 BFLAG Mode Select (MB0, MB1)** – Bits 6 and 7 select the mode of operation as follows:

MB1	MB0	BFLAG Function
0	0	Logic 0 output
0	1	Logic 1 output
1	0	IBF flag output (CSR.0)
1	1	Port enable (PE)

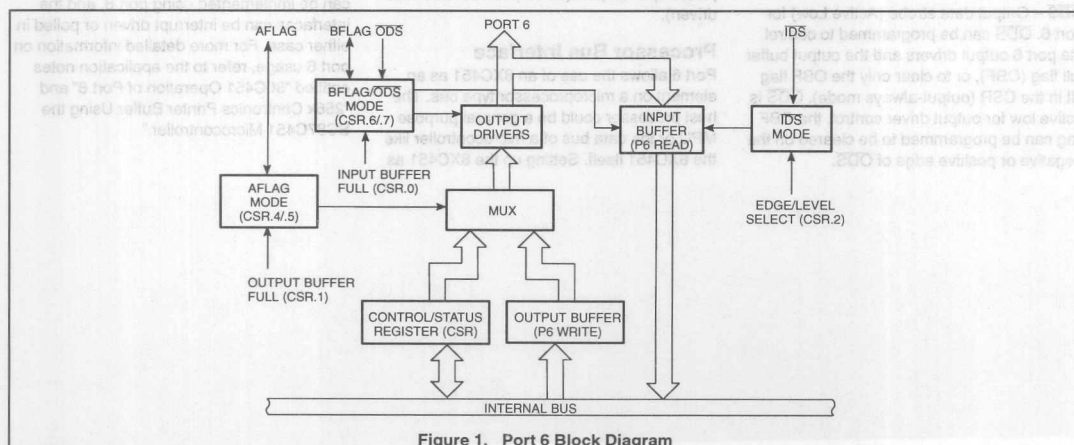
In the port enable mode,  $\overline{IDS}$  and  $\overline{ODS}$  inputs are disabled when BFLAG input is high. When the BFLAG input is low, the port is enabled for I/O.

**SPECIAL FUNCTION REGISTER ADDRESSES**

The SFRs are identical to those of the standard 80C51 with the exception of four registers that have been added to allow control of the three additional I/O ports P4, P5, and P6. The additional registers are P4, P5, P6, and CSR. Registers P4, P5, and P6 function as port latches for ports 4, 5, and 6, respectively. These registers operate identically to those for ports 0 through 3 of the 80C51.

**Table 1. Special Function Register Addresses**

REGISTER ADDRESS			BIT ADDRESS							
NAME	SYMBOL	ADDRESS	MSB							LSB
Port 4	P4	C0	C7	C6	C5	C4	C3	C2	C1	C0
Port 5	P5	C8	CF	CE	CD	CC	CB	CA	C9	C8
Port 6 data	P6	D8	DF	DE	DD	DC	DB	DA	D9	D8
Port 6 control status	CSR	E8	EF	EE	ED	EC	EB	EA	E9	E8

**Figure 1. Port 6 Block Diagram**

## CMOS single-chip 8-bit microcontroller 80C451/83C451/87C451

Table 2. 8X451 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT NAMES AND ADDRESSES								RESET VALUE
			MSB				LSB				
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
			EF	EE	ED	EC	EB	EA	E9	E8	
CSR#	Port 6 command/status	E8H	MB1	MB0	MA1	MA0	OBFC	IDSM	OBF	IBF	FCH
DPTR	Data pointer (2 bytes)										
DPH	Data pointer high	83H									00H
DPL	Data pointer low	82H									00H
IP*	Interrupt priority	B8H	BF	BE	BD	BC	BB	BA	B9	B8	xxx0000B
			-	-	-	PS	PT1	PX1	PT0	PX0	
IE*	Interrupt enable	A8H	AF	AE	AD	AC	AB	AA	A9	A8	0xx0000B
P0*	Port 0	80H	EA	-	-	ES	ET1	EX1	ET0	EX0	FFH
P1*	Port 1	90H	87	B6	85	84	83	82	81	80	FFH
P2*	Port 2	A0H	97	96	95	94	93	92	91	90	FFH
P3*	Port 3	B0H	A7	A6	A5	A4	A3	A2	A1	A0	FFH
P4*	Port 4	C0H	B7	B6	B5	B4	B3	B2	B1	B0	FFH
P5*	Port 5	C8H	C7	C6	C5	C4	C3	C2	C1	C0	FFH
P6*	Port 6	D8H	CF	CE	CD	CC	CB	CA	C9	C8	FFH
			DF	DE	DD	DC	DB	DA	D9	D8	FFH
PCON	Power control	87H	SMOD	-	-	-	GF1	GF0	PD	IDL	0xxx0000B
PSW*	Program status word	D0H	D7	D6	D5	D4	D3	D2	D1	D0	00H
SBUF	Serial data buffer	99H	CY	AC	F0	RS1	RS0	OV	-	P	xxxxxxxB
SCON*	Serial port control	98H	9F	9E	9D	9C	9B	9A	99	98	
SP	Stack pointer	81H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00H
TCON*	Timer/counter control	88H	8F	8E	8D	8C	8B	8A	89	88	07H
			TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
TMOD	Timer/counter mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H
TH0	Timer 0 high byte	8CH									00H
TH1	Timer 1 high byte	8DH									00H
TL0	Timer 0 low byte	8AH									00H
TL1	Timer 1 low byte	8BH									00H

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.

## CMOS single-chip 8-bit microcontroller 80C451/83C451/87C451

Table 3. Control Status Register (CSR)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MB1	MB0	MA1	MA0	OBFC	IDS <sub>M</sub>	OBF	IBF
BFLAG Mode Select		AFLAG Mode Select		Output Buffer Flag Clear Mode	Input Data Strobe Mode	Output Buffer Flag Full	Input Buffer Flag Full
0/0 = Logic 0 output* 0/1 = Logic 1 output* 1/0 = IBF output 1/1 = PE input (0 = Select) (1 = Disable I/O)		0/0 = Logic 0 output* 0/1 = Logic 1 output* 1/0 = OBF output 1/1 = SEL input (0 = Select) (1 = Control/status)		0 = Negative edge of ODS 1 = Positive edge o ODS	0 = Positive edge of IDS 1 = Low level of IDS	0 = Output data buffer empty 1 = Output data buffer full	0 = Input data buffer empty 1 = Input data buffer full

**NOTE:**

\* Output-always mode: MB1 = 0, MA1 = 1, and MA0 = 0. In this mode, port 6 is always enabled for output.  $\overline{\text{ODS}}$  only clears the OBF flag.

**ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>**

	BA	DA	AA	PARAMETER	CA	SA	FA		RATING	UNIT
Operating temperature under bias	T <sub>A3</sub>	T <sub>B3</sub>	T <sub>D3</sub>	-	-	A3	H9A		0 to +70 -40 to +85	°C
Storage temperature range	T <sub>S3</sub>	T <sub>B3</sub>	T <sub>D3</sub>	-	-	A3	H9C		-65 to +150	°C
Voltage on any other pin to V <sub>SS</sub>	V <sub>E3</sub>	V <sub>F3</sub>	V <sub>G3</sub>	-	-	A3	H9B		-0.5 to +6.5	V
Power dissipation (based on package heat transfer limitations, not device power consumption)	P <sub>D3</sub>	P <sub>B3</sub>	P <sub>D3</sub>	-	-	A3	H9D		1.5	W

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
2. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
3. Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to  $V_{SS}$  unless otherwise noted.

## CMOS single-chip 8-bit microcontroller 80C451/83C451/87C451

DC ELECTRICAL CHARACTERISTICS<sup>1</sup>T<sub>amb</sub> = 0°C to +70°C or -40°C to +85°C, V<sub>CC</sub> = 5V ±10%, V<sub>SS</sub> = 0V (87C451, 83C451, 80C451)

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYPICAL <sup>1</sup>	MAX	
V <sub>IL</sub>	Input low voltage; except EA		-0.5		0.2V <sub>CC</sub> -0.1	V
V <sub>IL1</sub>	Input low voltage to EA		0		0.2V <sub>CC</sub> -0.3	V
V <sub>IH</sub>	Input high voltage; except XTAL1, RST		0.2V <sub>CC</sub> +0.9		V <sub>CC</sub> +0.5	V
V <sub>IH1</sub>	Input high voltage; XTAL1, RST		0.7V <sub>CC</sub>		V <sub>CC</sub> +0.5	V
V <sub>OL</sub>	Output low voltage; ports 1, 2, 3, 4, 5, 6	I <sub>OL</sub> = 1.6mA <sup>2</sup>			0.45	V
V <sub>OL1</sub>	Output low voltage; port 0, ALE, PSEN	I <sub>OL</sub> = 3.2mA <sup>2</sup>			0.45	V
V <sub>OH</sub>	Output high voltage; ports 1, 2, 3, 4, 5, 6	I <sub>OH</sub> = -60µA	2.4			V
		I <sub>OH</sub> = -25µA	0.75V <sub>CC</sub>			V
		I <sub>OH</sub> = -10µA	0.9V <sub>CC</sub>			V
V <sub>OH1</sub>	Output high voltage (port 0 in external bus mode, ALE, PSEN) <sup>3</sup>	I <sub>OH</sub> = -800µA	2.4			V
		I <sub>OH</sub> = -300µA	0.75V <sub>CC</sub>			V
		I <sub>OH</sub> = -80µA	0.9V <sub>CC</sub>			V
I <sub>IL</sub>	Logical 0 input current; ports 1, 2, 3, 4, 5, 6	V <sub>IN</sub> = 0.45V			-50	µA
I <sub>TL</sub>	Logical 1-to-0 transition current; ports 1, 2, 3, 4, 5, 6	See note 4			-650	µA
I <sub>LI</sub>	Input leakage current; port 0	V <sub>IN</sub> = V <sub>IL</sub> or V <sub>IH</sub>			±10	µA
I <sub>CC</sub>	Power supply current:	See note 6				
	Active mode @ 12MHz <sup>5</sup>			11.5	25	mA
	Idle mode @ 12MHz <sup>5</sup>			1.3	4	mA
	Power down mode			3	50	µA
R <sub>RST</sub>	Internal reset pull-down resistor		50		300	kΩ
C <sub>IO</sub>	Pin capacitance <sup>7</sup>				10	pF

## NOTES:

- Typical ratings are based on a limited number of samples taken from early manufacturing lots and are not guaranteed. The values listed are at room temperature, 5V.
- Capacitive loading on ports 0 and 2 may cause spurious noise to be superimposed on the V<sub>OL</sub>s of ALE and ports 1 and 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100pF), the noise pulse on the ALE pin may exceed 0.8V. In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.
- Capacitive loading on ports 0 and 2 may cause the V<sub>OH</sub> on ALE and PSEN to momentarily fall below the 0.9V<sub>CC</sub> specification when the address bits are stabilizing.
- Pins of ports 1, 2 and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when V<sub>IN</sub> is approximately 2V.
- I<sub>CCMAX</sub> at other frequencies is given by:  
Active mode: I<sub>CCMAX</sub> = 0.94 X FREQ + 13.71  
Idle mode: I<sub>CCMAX</sub> = 0.14 X FREQ + 2.31  
where FREQ is the external oscillator frequency in MHz. I<sub>CCMAX</sub> is given in mA. See Figure 13.
- See Figures 14 through 17 for I<sub>CC</sub> test conditions.
- C<sub>IO</sub> applies to ports 1 through 6, AFLAG, BFLAG, XTAL1, XTAL2.

8	100	100	100	100	100	100
8	100	100	100	100	100	100
8	100	100	100	100	100	100
8	100	100	100	100	100	100
8	100	100	100	100	100	100
8	100	100	100	100	100	100

NOTES: SEE NEXT PAGE

## CMOS single-chip 8-bit microcontroller 80C451/83C451/87C451

AC ELECTRICAL CHARACTERISTICS<sup>1</sup>T<sub>amb</sub> = 0°C to +70°C or -40°C to +85°C, V<sub>CC</sub> = 5V ±10%, V<sub>SS</sub> = 0V (87C451, 83C451, 80C451)<sup>2</sup>

SYMBOL	FIGURE	PARAMETER	12MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
1/f <sub>CLCL</sub>		Oscillator frequency: Speed Versions SC8XC451 C SC8XC451 G			3.5 3.5	12 16	MHz MHz
t <sub>LHLL</sub>	2	ALE pulse width	127		2t <sub>CLCL</sub> -40		ns
t <sub>AVLL</sub>	2	Address valid to ALE low	28		t <sub>CLCL</sub> -55		ns
t <sub>LLAX</sub>	2	Address hold after ALE low	48		t <sub>CLCL</sub> -35		ns
t <sub>LIV</sub>	2	ALE low to valid instruction in		234		4t <sub>CLCL</sub> -100	ns
t <sub>LLPL</sub>	2	ALE low to PSEN low	43		t <sub>CLCL</sub> -40		ns
t <sub>PLPH</sub>	2	PSEN pulse width	205		3t <sub>CLCL</sub> -45		ns
t <sub>PLIV</sub>	2	PSEN low to valid instruction in		145		3t <sub>CLCL</sub> -105	ns
t <sub>PXIX</sub>	2	Input instruction hold after PSEN	0		0		ns
t <sub>PXIZ</sub>	2	Input instruction float after PSEN		59		t <sub>CLCL</sub> -25	ns
t <sub>AVIV</sub>	2	Address to valid instruction in		312		5t <sub>CLCL</sub> -105	ns
t <sub>PLAZ</sub>	2	PSEN low to address float		10		10	ns
Data Memory							
t <sub>RLRH</sub>	3, 4	RD pulse width	400		6t <sub>CLCL</sub> -100		ns
t <sub>WLWH</sub>	3, 4	WR pulse width	400		6t <sub>CLCL</sub> -100		ns
t <sub>RLDV</sub>	3, 4	RD low to valid data in		252		5t <sub>CLCL</sub> -165	ns
t <sub>RHDX</sub>	3, 4	Data hold after RD	0		0		ns
t <sub>RHDZ</sub>	3, 4	Data float after RD		97		2t <sub>CLCL</sub> -70	ns
t <sub>LLDV</sub>	3, 4	ALE low to valid data in		517		8t <sub>CLCL</sub> -150	ns
t <sub>AVDV</sub>	3, 4	Address to valid data in		585		9t <sub>CLCL</sub> -165	ns
t <sub>LLWL</sub>	3, 4	ALE low to RD or WR low	200	300	3t <sub>CLCL</sub> -50	3t <sub>CLCL</sub> +50	ns
t <sub>AVWL</sub>	3, 4	Address valid to WR low or RD low	203		4t <sub>CLCL</sub> -130		ns
t <sub>QVWX</sub>	3, 4	Data valid to WR transition	23		t <sub>CLCL</sub> -60		ns
t <sub>WHQX</sub>	3, 4	Data hold after WR	33		t <sub>CLCL</sub> -50		ns
t <sub>RLAZ</sub>	3, 4	RD low to address float		0	0		ns
t <sub>WHLH</sub>	3, 4	RD or WR high to ALE high	43	123	t <sub>CLCL</sub> -40	t <sub>CLCL</sub> +40	ns
Shift Register							
t <sub>XLXL</sub>	5	Serial port clock cycle time	1.0		12t <sub>CLCL</sub>		μs
t <sub>QVXH</sub>	5	Output data setup to clock rising edge	700		10t <sub>CLCL</sub> -133		ns
t <sub>XHQX</sub>	5	Output data hold after clock rising edge	50		2t <sub>CLCL</sub> -117		ns
t <sub>XHDX</sub>	5	Input data hold after clock rising edge	0		0		ns
t <sub>XHDV</sub>	5	Clock rising edge to input data valid		700		10t <sub>CLCL</sub> -133	ns

NOTES: SEE NEXT PAGE



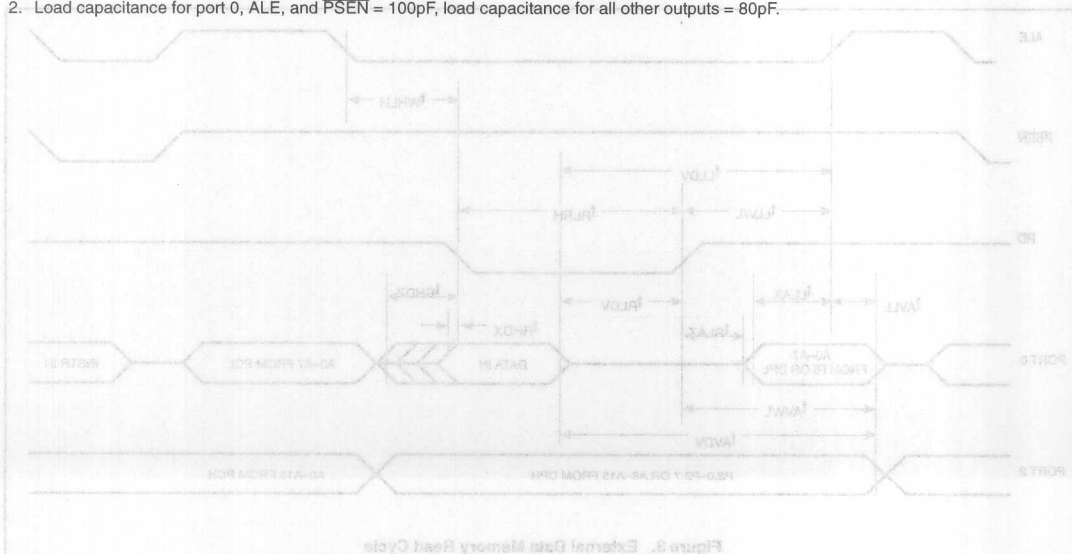
## CMOS single-chip 8-bit microcontroller 80C451/83C451/87C451

AC ELECTRICAL CHARACTERISTICS<sup>1</sup>T<sub>amb</sub> = 0°C to +70°C or -40°C to +85°C, V<sub>CC</sub> = 5V ±10%, V<sub>SS</sub> = 0V (87C451, 83C451, 80C451)<sup>2</sup>

SYMBOL	FIGURE	PARAMETER	12MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
Port 6 input (input rise and fall times = 5ns)							
t <sub>FLFH</sub>	8	PE width	270		3t <sub>CLCL</sub> +20		ns
t <sub>LIH</sub>	8	IDS width	270		3t <sub>CLCL</sub> +20		ns
t <sub>DVIH</sub>	8	Data setup to IDS high or PE high	0		0		ns
t <sub>HDX</sub>	8	Data hold after IDS high or PE high	30		30		ns
t <sub>VFV</sub>	9	IDS to BFLAG (IBF) delay		130		130	ns
Port 6 output							
t <sub>LOH</sub>	6	ODS width	270		3t <sub>CLCL</sub> +20		ns
t <sub>FVDV</sub>	7	SEL to data out delay		85		85	ns
t <sub>OLDV</sub>	6	ODS to data out delay		80		80	ns
t <sub>OHZ</sub>	6	ODS to data float delay		35		35	ns
t <sub>OVFV</sub>	6	ODS to AFLAG (OBF) delay		100		100	ns
t <sub>FLDV</sub>	6	PE to data out delay		120		120	ns
t <sub>OHFH</sub>	7	ODS to AFLAG (SEL) delay	100		100		ns
External Clock							
t <sub>CHCX</sub>	10	High time	20		20		ns
t <sub>CLCX</sub>	10	Low time	20		20		ns
t <sub>CLCH</sub>	10	Rise time		20		20	ns
t <sub>CHCL</sub>	10	Fall time		20		20	ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.



## CMOS single-chip 8-bit microcontroller 80C451/83C451/87C451

## EXPLANATION OF THE AC SYMBOLS

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

A – Address

C – Clock

D – Input data

H – Logic level high

I – Instruction (program memory contents)

L – Logic level low, or ALE

P – PSEN

Q – Output data

R – RD signal

t – Time

V – Valid

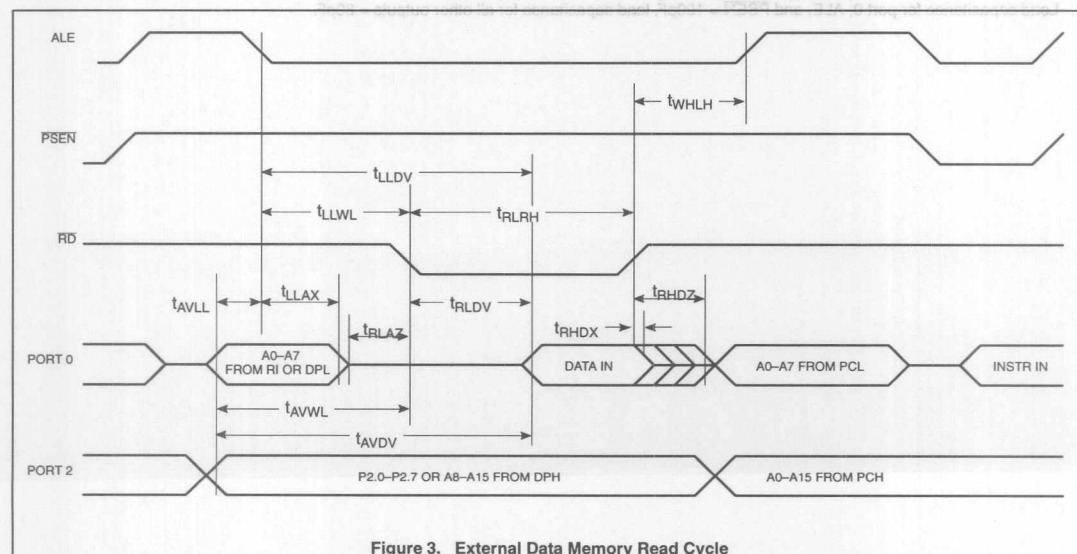
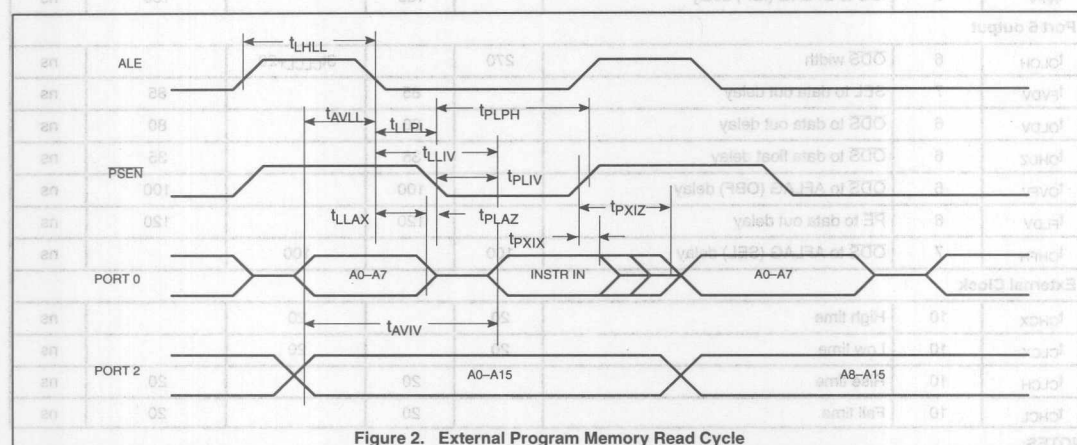
W – WR signal

X – No longer a valid logic level

Z – Float

Examples:  $t_{AVLL}$  = Time for address valid to ALE low.

$t_{LLPL}$  = Time for ALE low to PSEN low.



## CMOS single-chip 8-bit microcontroller 80C451/83C451/87C451

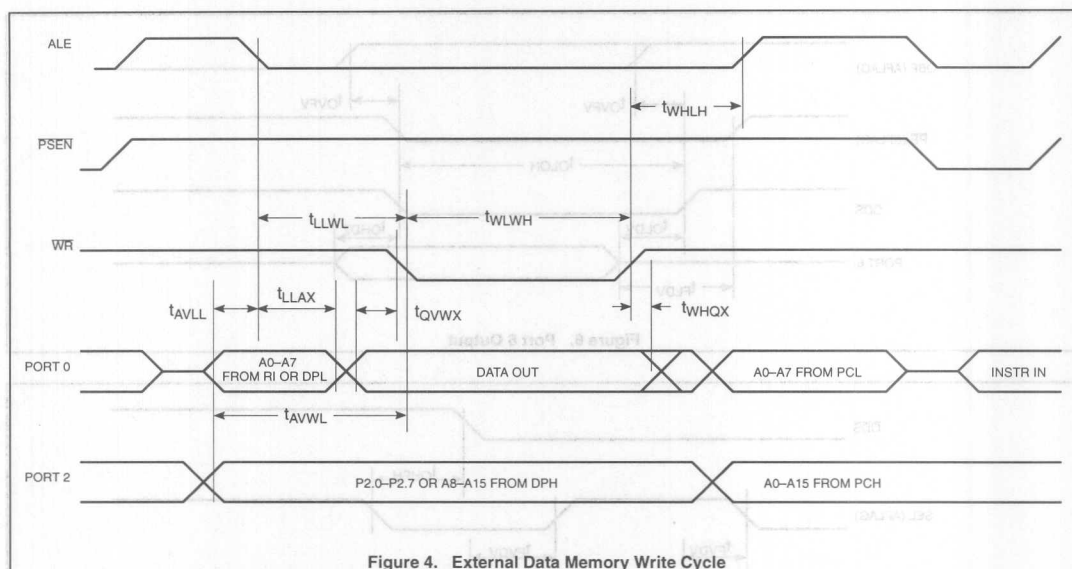


Figure 4. External Data Memory Write Cycle

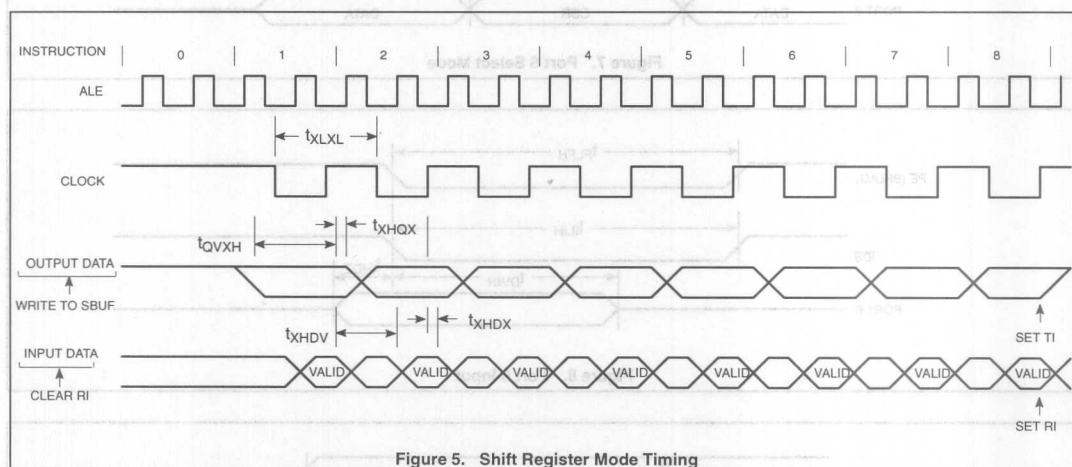
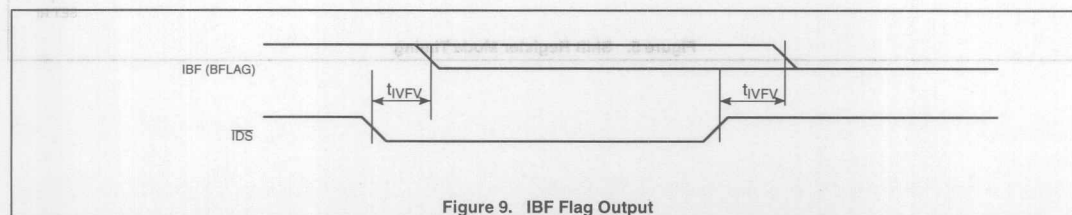
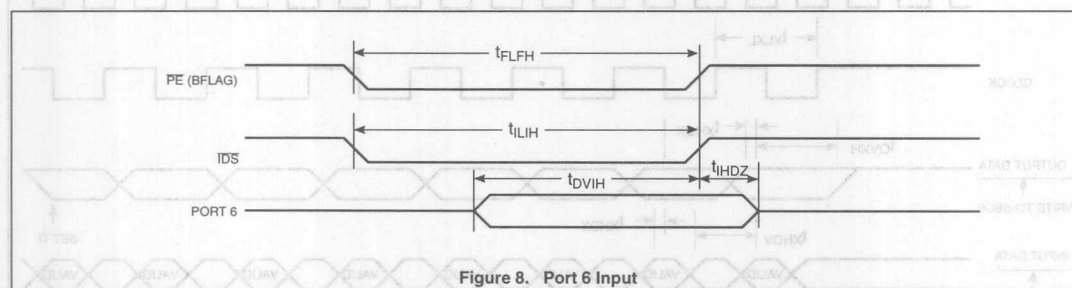
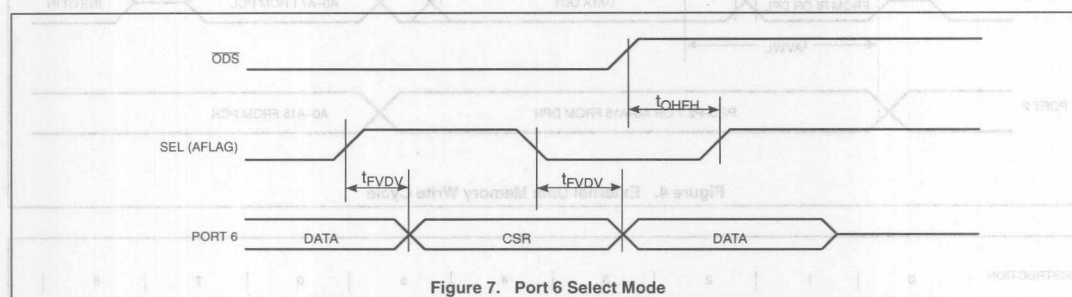
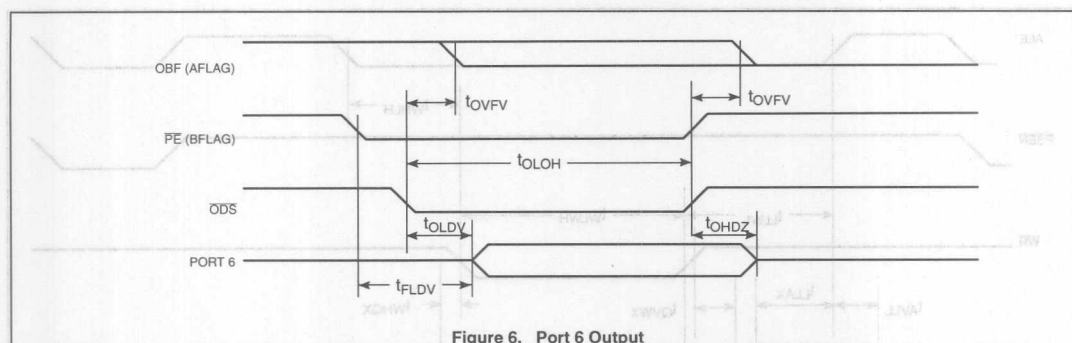


Figure 5. Shift Register Mode Timing

## CMOS single-chip 8-bit microcontroller 80C451/83C451/87C451



## CMOS single-chip 8-bit microcontroller 80C451/83C451/87C451

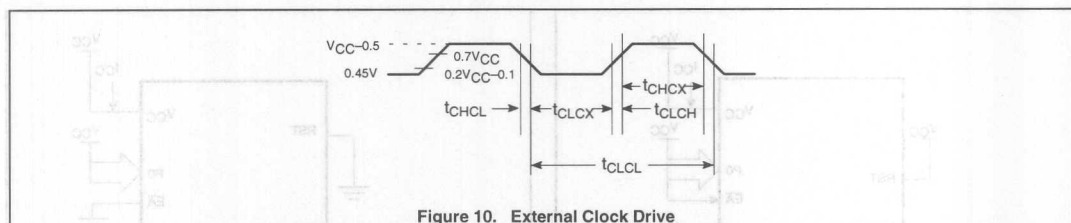


Figure 10. External Clock Drive

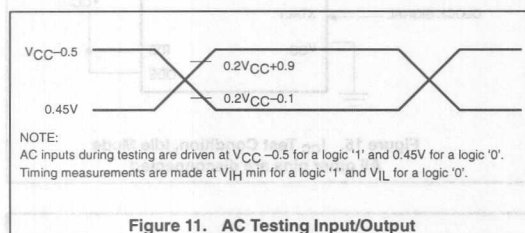


Figure 11. AC Testing Input/Output

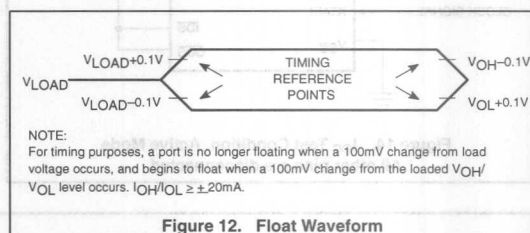
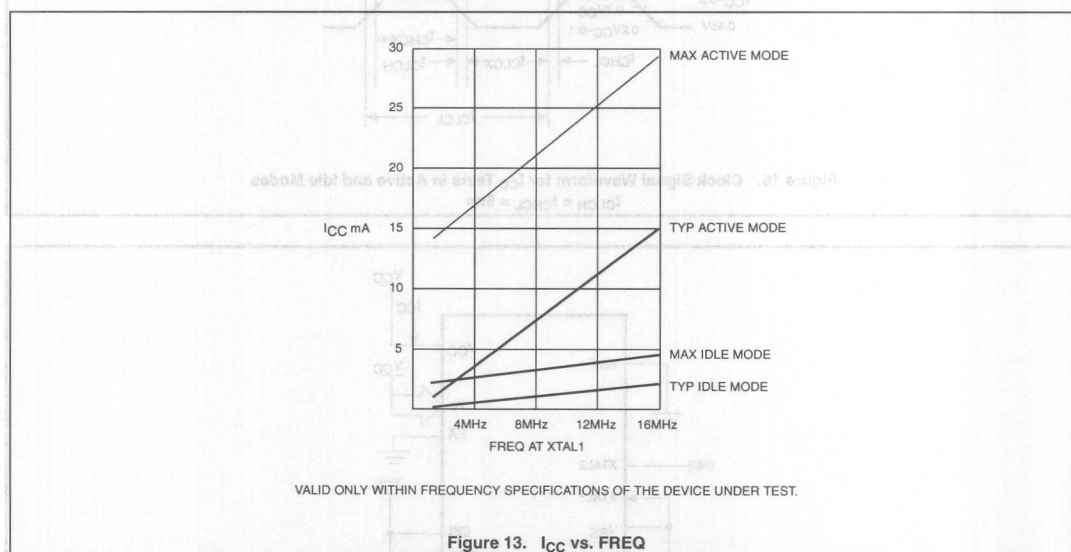
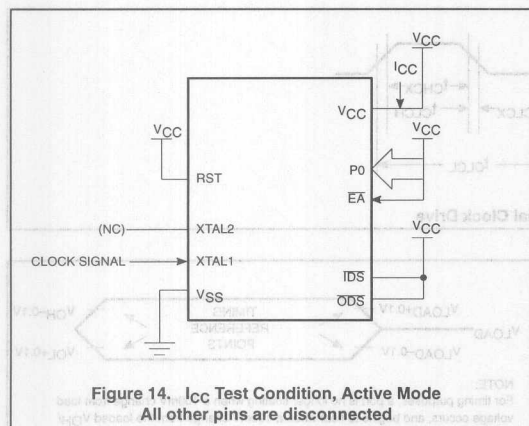


Figure 12. Float Waveform

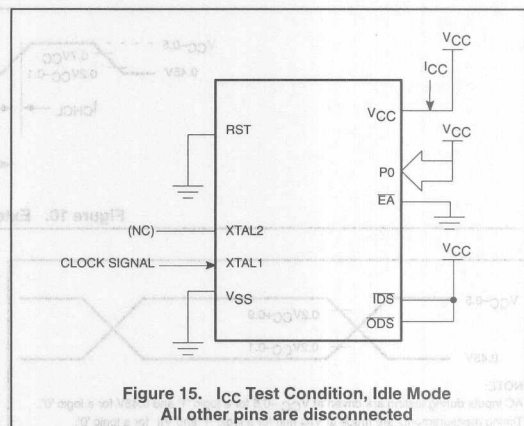
Figure 13.  $I_{CC}$  vs. FREQ



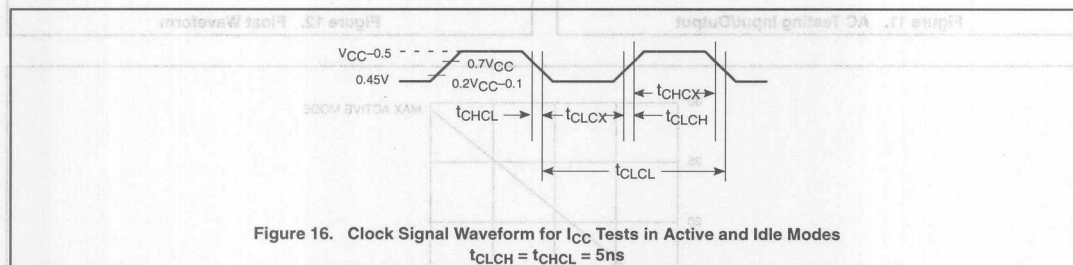
## CMOS single-chip 8-bit microcontroller 80C451/83C451/87C451



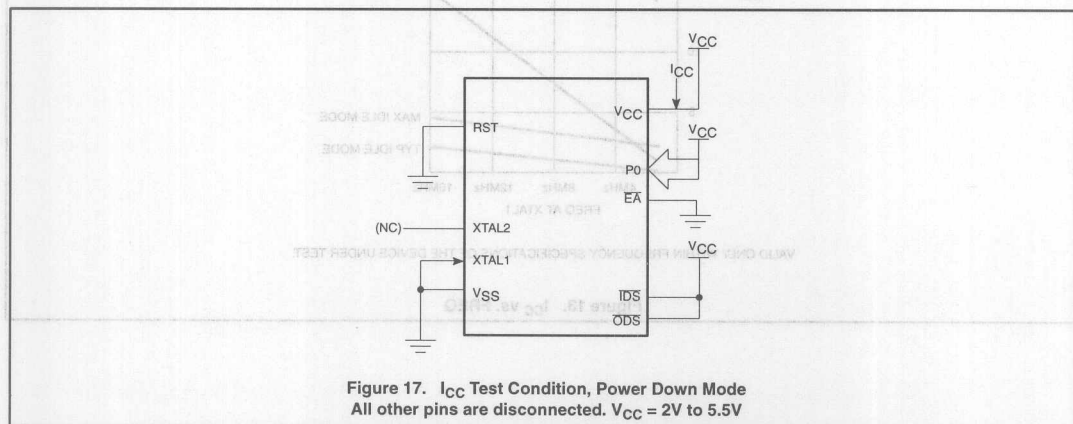
**Figure 14. I<sub>CC</sub> Test Condition, Active Mode**  
All other pins are disconnected



**Figure 15. I<sub>CC</sub> Test Condition, Idle Mode**  
All other pins are disconnected



**Figure 16. Clock Signal Waveform for I<sub>CC</sub> Tests in Active and Idle Modes**  
 $t_{CLCH} = t_{CHCL} = 5\text{ns}$



**Figure 17. I<sub>CC</sub> Test Condition, Power Down Mode**  
All other pins are disconnected. V<sub>CC</sub> = 2V to 5.5V

## CMOS single-chip 8-bit microcontroller 80C451/83C451/87C451

**EPROM CHARACTERISTICS**

The 87C451 is programmed by using a modified Quick-Pulse Programming™ algorithm. It differs from older methods in the value used for  $V_{PP}$  (programming supply voltage) and in the width and number of the ALE/PROG pulses.

The 87C451 contains two signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes identify the device as an 87C451 manufactured by Philips Semiconductors.

Table 4 shows the logic levels for reading the signature byte, and for programming the program memory, the encryption table, and the lock bits. The circuit configuration and waveforms for quick-pulse programming are shown in Figures 18 and 19. Figure 20 shows the circuit configuration for normal program memory verification.

**Quick-Pulse Programming**

The setup for microcontroller quick-pulse programming is shown in Figure 18. Note that the 87C451 is running with a 4 to 6MHz oscillator. The reason the oscillator needs to be running is that the device is executing internal address and program data transfers.

The address of the EPROM location to be programmed is applied to ports 1 and 2, as shown in Figure 18. The code byte to be programmed into that location is applied to port 0. RST, PSEN and pins of ports 2 and 3 specified in Table 4 are held at the 'Program Code Data' levels indicated in Table 4. The ALE/PROG is pulsed low 25 times as shown in Figure 19.

To program the encryption table, repeat the 25 pulse programming sequence for addresses 0 through 1FH, using the 'Pgm Encryption Table' levels. Do not forget that after the encryption table is programmed, verification cycles will produce only encrypted data.

To program the lock bits, repeat the 25 pulse programming sequence using the 'Pgm Lock Bit' levels. After one lock bit is programmed, further programming of the code memory and encryption table is disabled. However, the other lock bit can still be programmed.

Note that the  $\overline{EA}/V_{PP}$  pin must not be allowed to go above the maximum specified  $V_{PP}$  level for any amount of time. Even a narrow glitch above that voltage can cause permanent damage to the device. The  $V_{PP}$  source should be well regulated and free of glitches and overshoot.

**Program Verification**

If lock bit 2 has not been programmed, the on-chip program memory can be read out for program verification. The address of the program memory locations to be read is applied to ports 1 and 2 as shown in Figure 20. The other pins are held at the 'Verify Code Data' levels indicated in Table 4. The contents of the address location will be emitted on port 0. External pull-ups are required on port 0 for this operation.

If the encryption table has been programmed, the data presented at port 0 will be the exclusive NOR of the program byte with one of the encryption bytes. The user will have to know the encryption table contents in order to correctly decode the verification data. The encryption table itself cannot be read out.

**Reading the Signature Bytes**

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. The values are:

(030H) = 15H indicates manufactured by Philips

(031H) = 90H indicates 87C451

**Program/Verify Algorithms**

Any algorithm in agreement with the conditions listed in Table 4, and which satisfies the timing specifications, is suitable.

**Erase Characteristics**

Erase of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent erasure. **For this and secondary effects, it is recommended that an opaque label be placed over the window.** For elevated temperature or environments where solvents are being used, apply Kapton tape Fluorglas part number 2345-5, or equivalent.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 angstroms) to an integrated dose of at least 15W-sec/cm<sup>2</sup>. Exposing the EPROM to an ultraviolet lamp of 12,000µW/cm<sup>2</sup> rating for 20 to 39 minutes, at a distance of about 1 inch, should be sufficient.

Erase leaves the array in an all 1s state.

**Table 4. EPROM Programming Modes**

MODE	RST	PSEN	ALE/PROG	$\overline{EA}/V_{PP}$	P2.7	P2.6	P3.7	P3.6
Read signature	1	0	1	1	0	0	0	0
Program code data	1	0	0*	$V_{PP}$	1	0	1	1
Verify code data	1	0	1	1	0	0	1	1
Pgm encryption table	1	0	0*	$V_{PP}$	1	0	1	0
Pgm lock bit 1	1	0	0*	$V_{PP}$	1	1	1	1
Pgm lock bit 2	1	0	0*	$V_{PP}$	1	1	0	0

**NOTES:**

1. '0' = Valid low for that pin, '1' = valid high for that pin.

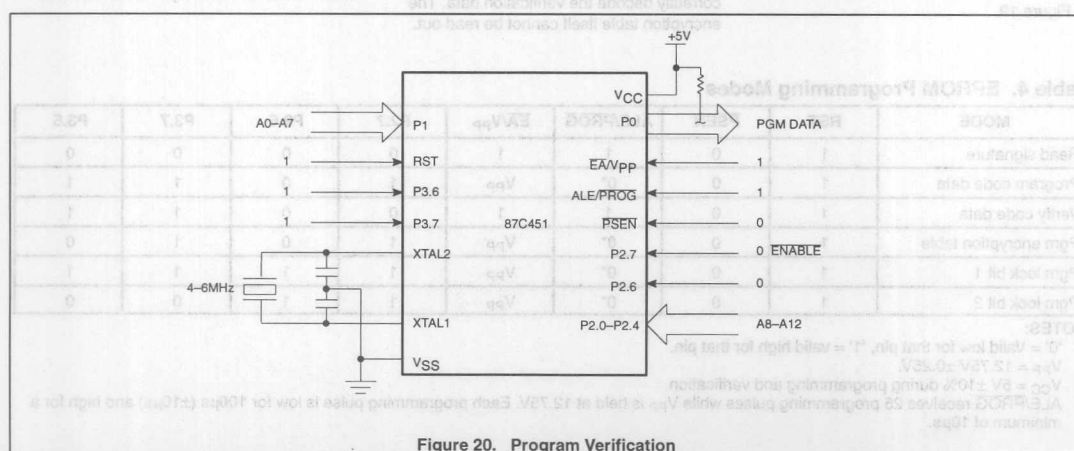
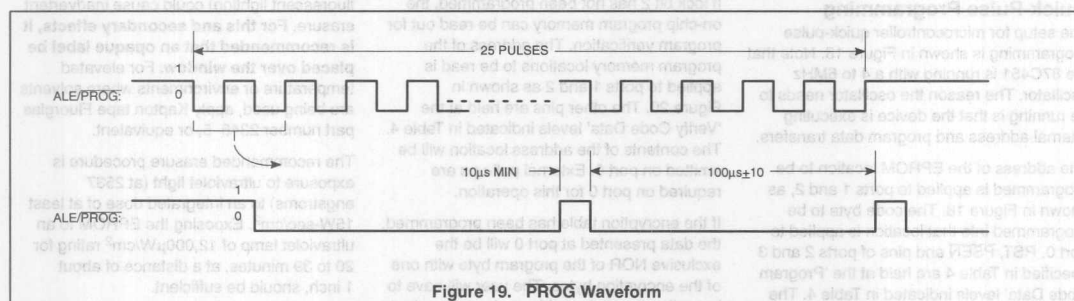
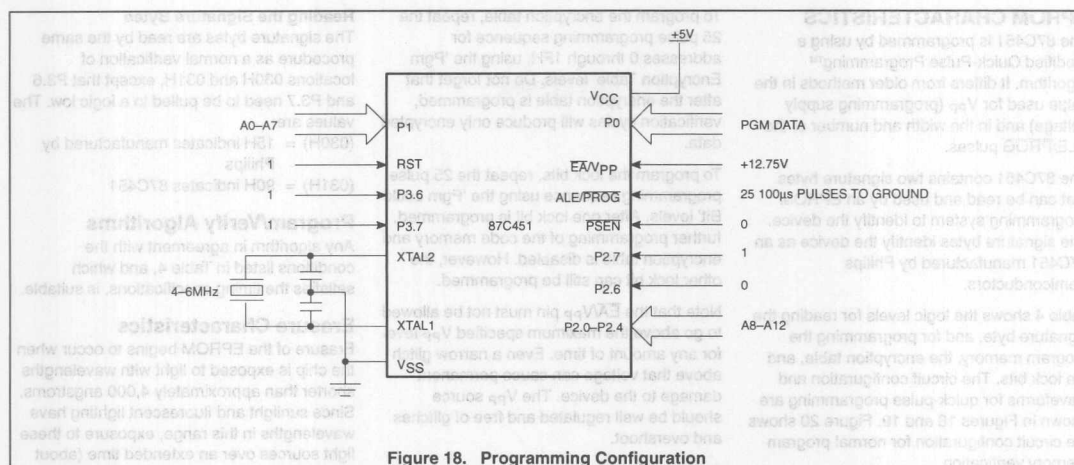
2.  $V_{PP}$  = 12.75V  $\pm$ 0.25V.

3.  $V_{CC}$  = 5V  $\pm$ 10% during programming and verification.

\* ALE/PROG receives 25 programming pulses while  $V_{PP}$  is held at 12.75V. Each programming pulse is low for 100µs ( $\pm$ 10µs) and high for a minimum of 10µs.

™Trademark phrase of Intel Corporation.

## CMOS single-chip 8-bit microcontroller 80C451/83C451/87C451



## CMOS single-chip 8-bit microcontroller 80C451/83C451/87C451

## EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

$T_{amb} = 21^{\circ}\text{C}$  to  $+27^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$  (See Figure 21)

SYMBOL	PARAMETER	MIN	MAX	UNIT
$V_{PP}$	Programming supply voltage	12.5	13.0	V
$I_{PP}$	Programming supply current		50	mA
$1/t_{CLCL}$	Oscillator frequency	4	6	MHz
$t_{AVGL}$	Address setup to PROG low	$48t_{CLCL}$		
$t_{GHAX}$	Address hold after PROG	$48t_{CLCL}$		
$t_{DVGL}$	Data setup to PROG low	$48t_{CLCL}$		
$t_{GHDX}$	Data hold after PROG	$48t_{CLCL}$		
$t_{ESH}$	P2.7 (ENABLE) high to $V_{PP}$	$48t_{CLCL}$		
$t_{SHGL}$	$V_{PP}$ setup to PROG low	10		$\mu\text{s}$
$t_{GHSL}$	$V_{PP}$ hold after PROG	10		$\mu\text{s}$
$t_{GLGH}$	PROG width	90	110	$\mu\text{s}$
$t_{AVQV}$	Address to data valid		$48t_{CLCL}$	
$t_{ELOZ}$	ENABLE low to data valid		$48t_{CLCL}$	
$t_{EHQZ}$	Data float after ENABLE	0	$48t_{CLCL}$	
$t_{GHGL}$	PROG high to PROG low	10		$\mu\text{s}$

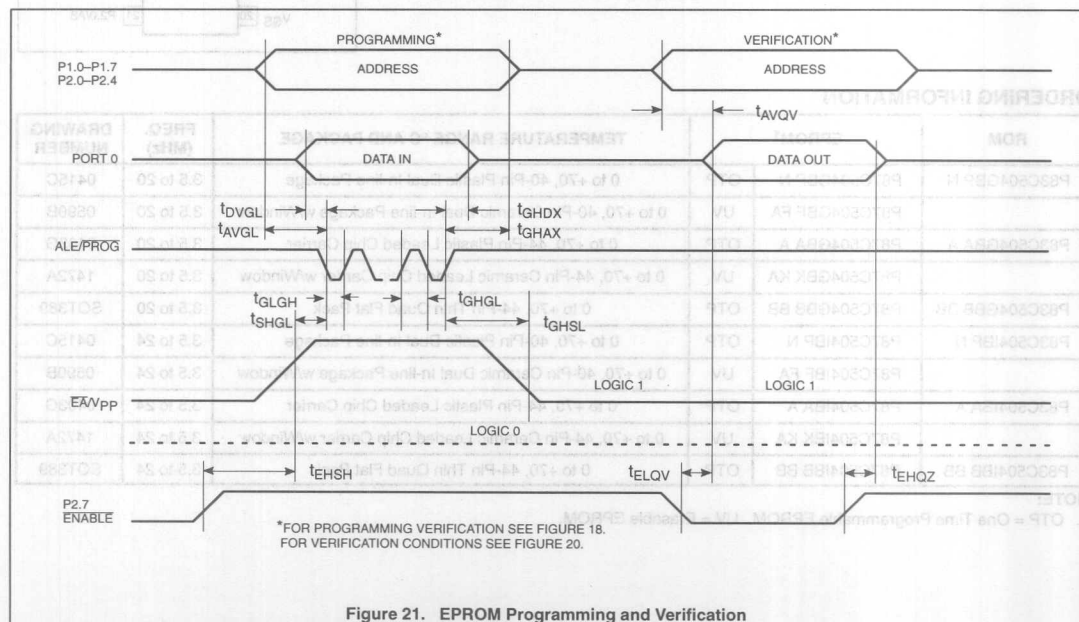


Figure 21. EPROM Programming and Verification

# CMOS single-chip 8-bit microcontroller 83C504/87C504

## DESCRIPTION

The 83C504 and 87C504 (hereafter referred to as 8XC504) Single-Chip 8-Bit Microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 83C504 has the same instruction set as the 80C51.

This device provides architectural enhancements that make it applicable in a variety of applications for general control systems. The 8XC504 contains 16k x 8 EPROM memory, the 83C504 contains 16k x 8 ROM memory, a volatile 256 x 8 read/write data memory, four 8-bit I/O ports, two 16-bit timer/event counters, a multi-source, two-priority-level, nested interrupt structure, a 24-by-8 bit unsigned divide, an enhanced UART and on-chip oscillator and timing circuits. For systems that require extra capability, the 8XC504 can be expanded using standard TTL compatible memories and logic.

## FEATURES

- 80C51 central processing unit
- 16k x 8 EPROM expandable externally to 64k bytes
  - Quick Pulse programming algorithm
  - Two level program security system
- 256 x 8 RAM, expandable externally to 64k bytes
- Two 16-bit timer/counters
- Four 8-bit I/O ports
- Full-duplex enhanced UART
  - Framing error detection
  - Automatic address recognition
- Power control modes
  - Idle mode
  - Power-down mode
- Once (On Circuit Emulation) Mode
- Five package styles
- OTP package available
- 24-by-8 bit divide
  - Requires 8 machine cycles
  - 24-bit quotient and 8-bit remainder

## PIN CONFIGURATIONS

P1.0	1	40	VCC
P1.1	2	39	P0.0/AD0
P1.2	3	38	P0.1/AD1
P1.3	4	37	P0.2/AD2
P1.4	5	36	P0.3/AD3
P1.5	6	35	P0.4/AD4
P1.6	7	34	P0.5/AD5
P1.7	8	33	P0.6/AD6
RST	9	32	P0.7/AD7
RxD/P3.0	10	31	EA/Vpp
TxD/P3.1	11	30	ALE/PROG
INT0/P3.2	12	29	PSEN
INT1/P3.3	13	28	P2.7/A15
T0/P3.4	14	27	P2.6/A14
T1/P3.5	15	26	P2.5/A13
WR/P3.6	16	25	P2.4/A12
RD/P3.7	17	24	P2.3/A11
XTAL2	18	23	P2.2/A10
XTAL1	19	22	P2.1/A9
VSS	20	21	P2.0/A8

## ORDERING INFORMATION

ROM	EPROM <sup>1</sup>		TEMPERATURE RANGE °C AND PACKAGE	FREQ. (MHz)	DRAWING NUMBER
P83C504GBP N	P87C504GBP N	OTP	0 to +70, 40-Pin Plastic Dual In-line Package	3.5 to 20	0415C
	P87C504GBF FA	UV	0 to +70, 40-Pin Ceramic Dual In-line Package w/Window	3.5 to 20	0590B
P83C504GBA A	P87C504GBA A	OTP	0 to +70, 44-Pin Plastic Leaded Chip Carrier	3.5 to 20	0403G
	P87C504GBK KA	UV	0 to +70, 44-Pin Ceramic Leaded Chip Carrier w/Window	3.5 to 20	1472A
P83C504GBB BB	P87C504GBB BB	OTP	0 to +70, 44-Pin Thin Quad Flat Pack	3.5 to 20	SOT389
P83C504IBP N	P87C504IBP N	OTP	0 to +70, 40-Pin Plastic Dual In-line Package	3.5 to 24	0415C
	P87C504IBF FA	UV	0 to +70, 40-Pin Ceramic Dual In-line Package w/Window	3.5 to 24	0590B
P83C504IBA A	P87C504IBA A	OTP	0 to +70, 44-Pin Plastic Leaded Chip Carrier	3.5 to 24	0403G
	P87C504IBK KA	UV	0 to +70, 44-Pin Ceramic Leaded Chip Carrier w/Window	3.5 to 24	1472A
P83C504IBB BB	P87C504IBB BB	OTP	0 to +70, 44-Pin Thin Quad Flat Pack	3.5 to 24	SOT389

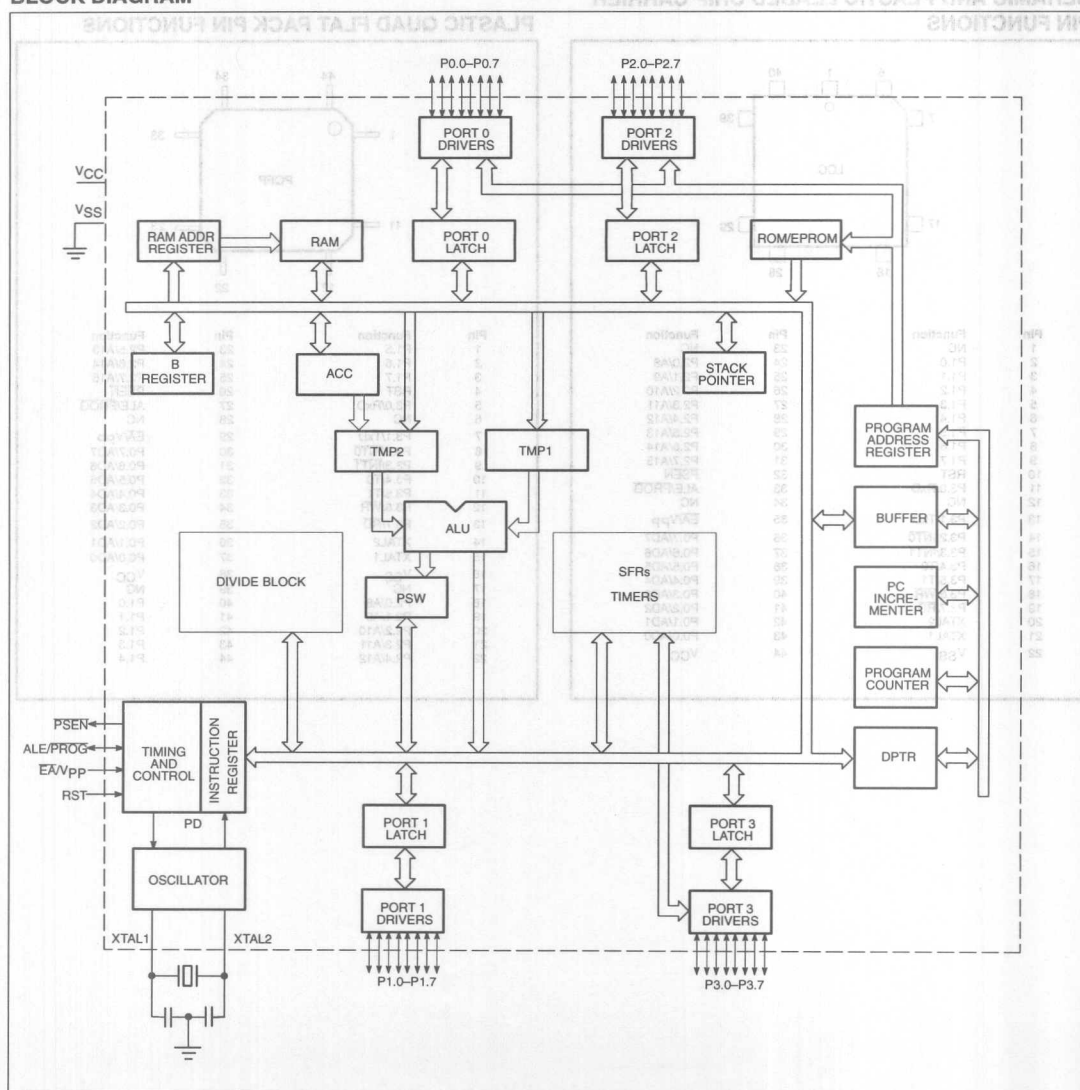
### NOTE:

1. OTP = One Time Programmable EPROM. UV = Erasable EPROM.



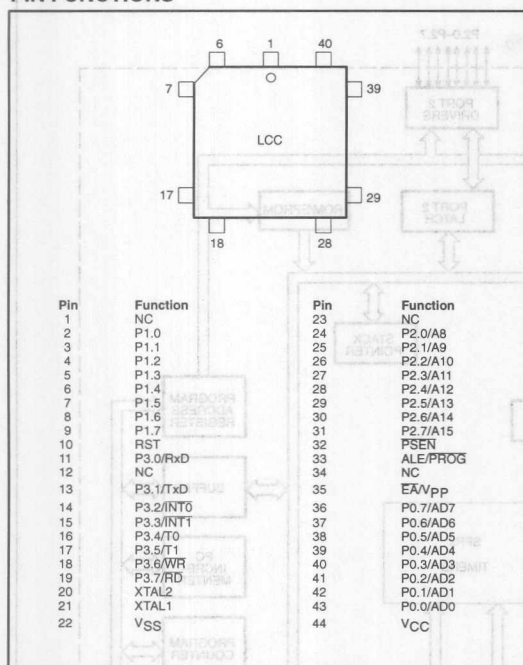
## CMOS single-chip 8-bit microcontroller 83C504/87C504

## BLOCK DIAGRAM

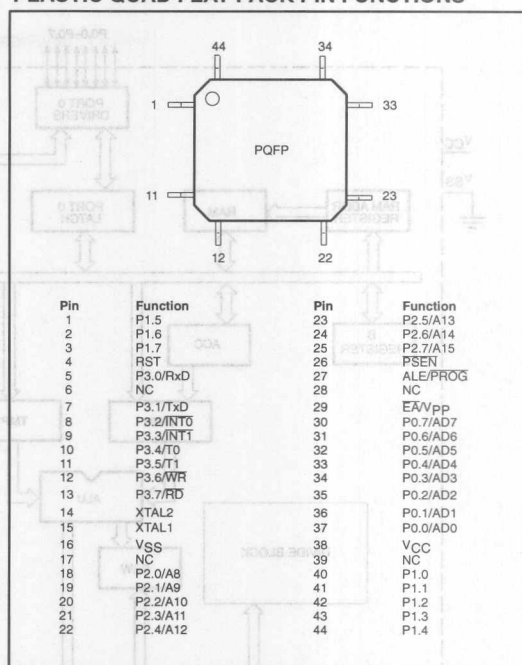


## CMOS single-chip 8-bit microcontroller

83C504/87C504

CERAMIC AND PLASTIC LEADED CHIP CARRIER  
PIN FUNCTIONS

## PLASTIC QUAD FLAT PACK PIN FUNCTIONS



## CMOS single-chip 8-bit microcontroller 83C504/87C504

## PIN DESCRIPTIONS

SYMBOL	DESCRIPTION	ADDRESS	DATA	BIT ADDRESS, SYMBOL OR ALTERNATE	PIN NUMBER			TYPE	NAME AND FUNCTION
					DIP	LCC	QFP		
V <sub>SS</sub>					20	22	16	I	Ground: 0V reference.
V <sub>CC</sub>					40	44	38	I	<b>Power Supply:</b> This is the power supply voltage for normal, idle, and power-down operation.
P0.0–0.7					39–32	43–36	37–30	I/O	<b>Port 0:</b> Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s. Port 0 also outputs the code bytes during program verification and receives code bytes during EPROM programming. External pull-ups are required during program verification.
P1.0–P1.7					1–8	2–9	40–44, 1–3	I/O	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pull-ups, except P1.6 and P1.7 which are open drain. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 1 also receives the low-order address byte during program memory verification.
P2.0–P2.7					21–28	24–31	18–25	I/O	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register. Some Port 2 pins receive the high order address bits during EPROM programming and verification.
P3.0–P3.7					10–17	11, 13–19	5, 7–13	I/O	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 3 also serves the special features of the 80C51 family, as listed below: <b>RxD (P3.0):</b> Serial input port <b>TxD (P3.1):</b> Serial output port <b>INT0 (P3.2):</b> External interrupt <b>INT1 (P3.3):</b> External interrupt <b>T0 (P3.4):</b> Timer 0 external input <b>T1 (P3.5):</b> Timer 1 external input <b>WR (P3.6):</b> External data memory write strobe <b>RD (P3.7):</b> External data memory read strobe
RST					9	10	4	I	<b>Reset:</b> A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to V <sub>SS</sub> permits a power-on reset using only an external capacitor to V <sub>CC</sub> .
ALE/PROG					30	33	27	I/O	<b>Address Latch Enable/Program Pulse:</b> Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory. This pin is also the program pulse input (PROG) during EPROM programming.
PSEN					29	32	26	O	<b>Program Store Enable:</b> The read strobe to external program memory. When the 83C504 is executing code from the external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory. PSEN is not activated during fetches from internal program memory.
E <sub>A</sub> V <sub>PP</sub>					31	35	29	I	<b>External Access Enable/Programming Supply Voltage:</b> E <sub>A</sub> must be externally held low to enable the device to fetch code from external program memory locations 0000H and 3FFFFH. If E <sub>A</sub> is held high, the device executes from internal program memory unless the program counter contains an address greater than 3FFFFH. This pin also receives the 12.75V programming supply voltage (V <sub>PP</sub> ) during EPROM programming. If security bit 1 is programmed, E <sub>A</sub> will be internally latched on Reset.
XTAL1					19	21	15	I	<b>Crystal 1:</b> Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2					18	20	14	O	<b>Crystal 2:</b> Output from the inverting oscillator amplifier.

## NOTE:

To avoid "latch-up" effect at power-on, the voltage on any pin at any time must not be higher than V<sub>CC</sub> + 0.5V or V<sub>SS</sub> – 0.5V, respectively.

## CMOS single-chip 8-bit microcontroller 87C504/87C504

Table 1. 87C04 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB						LSB		
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
AUXR#	Auxiliary	8EH	—	—	—	—	—	—	—	AO	xxxxxxx0B
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
DPTR:	Data Pointer (2 bytes)										
DPH	Data Pointer High	83H									00H
DPL	Data Pointer Low	82H									00H
			FF	FE	FD	FC	FB	FA	F9	F8	
DCON	Divide Control	FBH	—	—	—	—	—	AUTOD	DSTRT	DCTRL	00H
DVND0	Dividend LSB	91H									00H
DVND1	Dividend Middle Byte	92H									00H
DVND2	Dividend MSB	93H									00H
DVSR	Divisor	95H									00H
			AF	AE	AD	AC	AB	AA	A9	A8	
IE*	Interrupt Enable	A8H	EA	EC	ET2	ES	ET1	EX1	ET0	EX0	00H
			BF	BE	BD	BC	BB	BA	B9	B8	
IP*	Interrupt Priority	B8H	—	PPC	PT2	PS	PT1	PX1	PT0	PX0	x0000000B
			87	86	85	84	83	82	81	80	
P0*	Port 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
			97	96	95	94	93	92	91	90	
P1*	Port 1	90H	CEX4	CEX3	CEX2	CEX1	CEX0	EX1	T2EX	T2	FFH
			A7	A6	A5	A4	A3	A2	A1	A0	
P2*	Port 2	A0H	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8	FFH
			B7	B6	B5	B4	B3	B2	B1	B0	
P3*	Port 3	B0H	RD	WR	T1	T0	INT1	INT0	TxD	RxD	FFH
PCON	Power Control	87H	SMOD1	SMOD0	—	POF1	GF1	GF0	PD	IDL	00xxxx00B
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program Status Word	D0H	CY	AC	F0	RS1	RS0	OV	—	P	00H
RMDR	Remainder	94H									00H
SADDR#	Slave Address	A9H									00H
SADEN#	Slave Address Mask	B9H									00H
SBUF	Serial Data Buffer	99H									xxxxxxx0B
			9F	9E	9D	9C	9B	9A	99	98	
SCON*	Serial Control	98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00H
SP	Stack Pointer	81H									07H
			8F	8E	8D	8C	8B	8A	89	88	
TCON*	Timer Control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
TH0	Timer High 0	8CH									00H
TH1	Timer High 1	8DH									00H
TL0	Timer Low 0	8AH									00H
TL1	Timer Low 1	8BH									00H
TMOD	Timer Mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H
			C7	C6	C5	C4	C3	C2	C1	C0	

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.

1. Reset value depends on reset source.

## CMOS single-chip 8-bit microcontroller 83C504/87C504

**ENHANCED UART**

The 8XC504 UART has all of the capabilities of the standard 80C51 UART plus Framing Error Detection and Automatic Address Recognition. As in the 80C51, all four modes of operation are supported as well as the 9th bit in modes 2 and 3 that can be used to facilitate multiprocessor communication.

The Framing Error Detection allows the UART to look for missing stop bits. If a Stop bit is missing, the FE bit in the SCON SFR is set. The FE bit can be checked after each transmission to detect communication errors. The FE bit can only be cleared by software and is not affected by a valid stop bit.

Automatic Address Recognition is used to reduce the CPU service time for the serial port. The CPU only needs to service the UART when it is addressed and, with this done by the on-chip circuitry, the need for software overhead is greatly reduced. This mode works similar to the 9th bit communication mode, except that uses only 8 bits and the Stop bit is used to cause the RI bit to be set. There are two SFRs associated with this mode. They are SADDR, which holds the slave address and SADEN, which contains a mask that allows selective masking of the slave address so that broadcast addresses can be used.

**HARDWARE DIVIDE UNIT**

The 8XC504 contains a 24-by-8 bit hardware divide unit. The 24 bit dividend is stored in special function registers DVND0 (LSB) – DVND2 (MSB) and the divisor is in register DVSR. A division operation returns the 24-bit result in the dividend registers (DVND0 – DVND2) and an 8-bit remainder in register RMDR.

The divide unit provides two modes of operation, auto-start and flag-controlled. Auto-start mode is enabled by setting the AUTOD (auto divide) bit in the DCON (divide control) register. If auto-start mode is enabled, writing to the divisor register (DVSR) will automatically start a division operation and will set the DCTRL (divide control) and DSTRT bits in the DCON register. DCTRL will automatically be cleared by the divide hardware when the division operation has been completed.

Flag controlled operation is initiated by setting the DCTRL bit in the DCON register which will start the division operation and also set the DSTRT bit. The DCTRL bit will automatically be cleared by the divide hardware when the division operation has

been completed. DSTRT can only be cleared by software.

**POWER OFF FLAG**

The Power Off Flag (POF) is set by on-chip circuitry when the  $V_{CC}$  level on the 8XC504 rises from 0 to 5V. The POF bit can be set or cleared by software allowing a user to determine if the reset is the result of a power-on or a warm start after powerdown. The  $V_{CC}$  level must remain above 3V for the POF to remain unaffected by the  $V_{CC}$  level.

**OSCILLATOR CHARACTERISTICS**

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

**Reset**

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. To insure a good power-on reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few milliseconds) plus two machine cycles. At power-on, the voltage on  $V_{CC}$  and RST must come up at the same time for a proper start-up.

**Idle Mode**

In the idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

**Power-Down Mode**

To save even more power, a Power Down mode can be invoked by software. In this

mode, the oscillator is stopped and the instruction that invoked Power Down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power Down mode is terminated.

On the 8XC504 either a hardware reset or external interrupt can use an exit from Power Down. Reset redefines all the SFRs but does not change the on-chip RAM. An external interrupt allows both the SFRs and the on-chip RAM to retain their values.

To properly terminate Power Down the reset or external interrupt should not be executed before  $V_{CC}$  is restored to its normal operating level and must be held active long enough for the oscillator to restart and stabilize (normally less than 10ms).

With an external interrupt, INT0 and INT1 must be enabled and configured as level-sensitive. Holding the pin low restarts the oscillator but bringing the pin back high completes the exit. Once the interrupt is serviced, the next instruction to be executed after RETI will be the one following the instruction that put the device into Power Down.

**Design Consideration**

- When the idle mode is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

**ONCE™ Mode**

The ONCE ("On-Circuit Emulation") Mode facilitates testing and debugging of systems using the 8XC504 without the 8XC504 having to be removed from the circuit. The ONCE Mode is invoked by:

1. Pull ALE low while the device is in reset and PSEN is high;
2. Hold ALE low as RST is deactivated.

While the device is in ONCE Mode, the Port 0 pins go into a float state, and the other port pins and ALE and PSEN are weakly pulled high. The oscillator circuit remains active. While the 8XC504 is in this mode, an emulator or test CPU can be used to drive the circuit. Normal operation is restored when a normal reset is applied.



## CMOS single-chip 8-bit microcontroller 83C504/87C504

Table 2. External Pin Status During Idle and Power-Down Mode

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>

PARAMETER	RATING	UNIT
Operating temperature under bias	0 to +70 or -40 to +85	°C
Storage temperature range	-65 to +150	°C
Voltage on EAV <sub>PP</sub> pin to V <sub>SS</sub>	0 to +13.0	V
Voltage on any other pin to V <sub>SS</sub>	-0.5 to +6.5	V
Maximum I <sub>OL</sub> per I/O pin	15	mA
Power dissipation (based on package heat transfer limitations, not device power consumption)	1.5	W

## NOTES:

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
- This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
- Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V<sub>SS</sub> unless otherwise noted.

## Electrical Deviations from Commercial Specifications for Extended Temperature Range

DC and AC parameters not included here are the same as in the commercial temperature range table.

## DC ELECTRICAL CHARACTERISTICS

T<sub>amb</sub> = -40°C to +85°C, V<sub>CC</sub> = 5V ±10%, V<sub>SS</sub> = 0V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			MIN	MAX	
V <sub>IL</sub>	Input low voltage, except E <sub>A</sub>		-0.5	0.2V <sub>CC</sub> -0.15	V
V <sub>IL1</sub>	Input low voltage to E <sub>A</sub>		0	0.2V <sub>CC</sub> -0.35	V
V <sub>IH</sub>	Input high voltage, except XTAL1, RST		0.2V <sub>CC</sub> +1	V <sub>CC</sub> +0.5	V
V <sub>IH1</sub>	Input high voltage to XTAL1, RST		0.7V <sub>CC</sub> +0.1	V <sub>CC</sub> +0.5	V
I <sub>IL</sub>	Logical 0 input current, ports 1, 2, 3	V <sub>IN</sub> = 0.45V		-75	µA
I <sub>TL</sub>	Logical 1-to-0 transition current, ports 1, 2, 3	V <sub>IN</sub> = 2.0V		-750	µA
I <sub>CC</sub>	Power supply current: Active mode Idle mode Power-down mode	V <sub>CC</sub> = 4.5-5.5V, Frequency range = 3.5 to 24MHz		19 6 50	mA mA µA

## CMOS single-chip 8-bit microcontroller 83C504/87C504

## DC ELECTRICAL CHARACTERISTICS

 $T_{amb} = 0^{\circ}\text{C to } +70^{\circ}\text{C or } -40^{\circ}\text{C to } +85^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$ 

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP <sup>1</sup>	MAX	
$V_{IL}$	Input low voltage, except EA <sup>7</sup>		-0.5		$0.2V_{CC} - 0.1$	V
$V_{IL1}$	Input low voltage to EA <sup>7</sup>		0		$0.2V_{CC} - 0.3$	V
$V_{IH}$	Input high voltage, except XTAL1, RST <sup>7</sup>		$0.2V_{CC} + 0.9$		$V_{CC} + 0.5$	V
$V_{IH1}$	Input high voltage, XTAL1, RST <sup>7</sup>		$0.7V_{CC}$		$V_{CC} + 0.5$	V
$V_{OL}$	Output low voltage, ports 1, 2, 3 <sup>9</sup>	$I_{OL} = 100\mu\text{A}$ $I_{OL} = 1.6\text{mA}^2$ $I_{OL} = 3.5\text{mA}$			0.3 0.45 1.0	V V V
$V_{OL1}$	Output low voltage, port 0, ALE, PSEN <sup>9</sup>	$I_{OL} = 200\mu\text{A}$ $I_{OL} = 3.2\text{mA}^2$ $I_{OL} = 7.0\text{mA}$			0.3 0.45 1.0	V V V
$V_{OH}$	Output high voltage, ports 1, 2, 3, ALE, PSEN <sup>3</sup>	$I_{OH} = -60\mu\text{A}$ $I_{OH} = -30\mu\text{A}$ $I_{OH} = -10\mu\text{A}$	$V_{CC} - 1.5$ $V_{CC} - 0.7$ $V_{CC} - 0.3$			V V V
$V_{OH1}$	Output high voltage (port 0 in external bus mode), ALE <sup>10</sup> , PSEN <sup>3</sup>	$I_{OH} = -7.0\text{mA}$ $I_{OH} = -3.2\text{mA}$ $I_{OH} = -200\mu\text{A}$	$V_{CC} - 1.5$ $V_{CC} - 0.7$ $V_{CC} - 0.3$			V V V
$I_{IL}$	Logical 0 input current, ports 1, 2, 3 <sup>7</sup>	$V_{IN} = 0.45\text{V}$			-50	$\mu\text{A}$
$I_{TL}$	Logical 1-to-0 transition current, ports 1, 2, 3 <sup>7</sup>	See note 4			-650	$\mu\text{A}$
$I_{LI}$	Input leakage current, port 0	$0.45 V_{IN} < V_{CC} - 0.3$			$\pm 10$	$\mu\text{A}$
$I_{CC}$	Power supply current: 7 Active mode @ 24MHz <sup>5</sup> Idle mode @ 24MHz Power-down mode	See note 6		15 3 10	25 5 75	mA mA $\mu\text{A}$
$R_{RST}$	Internal reset pull-down resistor		50		225	k $\Omega$
$C_{IO}$	Pin capacitance <sup>11</sup> (except EA)				15	pF

## NOTES:

- Typical ratings are not guaranteed. The values listed are at room temperature, 5V.
- Capacitive loading on ports 0 and 2 may cause spurious noise to be superimposed on the  $V_{OL}$ s of ALE and ports 1 and 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100pF), the noise pulse on the ALE pin may exceed 0.8V. In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.  $I_{OL}$  can exceed these conditions provided that no single output sinks more than 5mA and no more than two outputs exceed the test conditions.
- Capacitive loading on ports 0 and 2 may cause the  $V_{OH}$  on ALE and PSEN to momentarily fall below the  $0.9V_{CC}$  specification when the address bits are stabilizing.
- Pins of ports 1, 2 and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when  $V_{IN}$  is approximately 2V.
- $I_{CCMAX}$  at other frequencies is given by: Active mode:  $I_{CCMAX} = 1.50 \times \text{FREQ} + 8$ ; Idle mode:  $I_{CCMAX} = 0.14 \times \text{FREQ} + 2.31$ , where FREQ is the external oscillator frequency in MHz.  $I_{CCMAX}$  is given in mA. See Figure 8.
- See Figures 9 through 12 for  $I_{CC}$  test conditions.
- These values apply only to  $T_{amb} = 0^{\circ}\text{C to } +70^{\circ}\text{C}$ . For  $T_{amb} = -40^{\circ}\text{C to } +85^{\circ}\text{C}$ , see table on previous page.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:  
Maximum  $I_{OL}$  per port pin: 15mA (\*NOTE: This is 85°C specification.)  
Maximum  $I_{OL}$  per 8-bit port: 26mA  
Maximum total  $I_{OL}$  for all outputs: 71mA  
If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
- ALE is tested to  $V_{OH1}$ , except when ALE is off then  $V_{OH}$  is the voltage specification.
- Pin capacitance is less than 25pF. Pin capacitance of ceramic package is less than 15pF (except EA is 25pF).

## CMOS single-chip 8-bit microcontroller 83C504/87C504

## AC ELECTRICAL CHARACTERISTICS

T<sub>amb</sub> = 0°C to +70°C or -40°C to +85°C, V<sub>CC</sub> = 5V ±10%, V<sub>SS</sub> = 0V<sup>1, 2, 3</sup>

SYMBOL	FIGURE	PARAMETER	24MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
t <sub>1/CLCL</sub>	1	Oscillator frequency			3.5	24	MHz
t <sub>LHLL</sub>	1	ALE pulse width	85		2t <sub>CLCL</sub> -40		ns
t <sub>AVLL</sub>	1	Address valid to ALE low	22		t <sub>CLCL</sub> -40		ns
t <sub>LLAX</sub>	1	Address hold after ALE low	32		t <sub>CLCL</sub> -30		ns
t <sub>LLIV</sub>	1	ALE low to valid instruction in		150		4t <sub>CLCL</sub> -100	ns
t <sub>LLPL</sub>	1	ALE low to PSEN low	32		t <sub>CLCL</sub> -30		ns
t <sub>PLPH</sub>	1	PSEN pulse width	142		3t <sub>CLCL</sub> -45		ns
t <sub>PLIV</sub>	1	PSEN low to valid instruction in		82		3t <sub>CLCL</sub> -105	ns
t <sub>PIXI</sub>	1	Input instruction hold after PSEN	0		0		ns
t <sub>PIXZ</sub>	1	Input instruction float after PSEN		37		t <sub>CLCL</sub> -25	ns
t <sub>AVIV</sub>	1	Address to valid instruction in		207		5t <sub>CLCL</sub> -105	ns
t <sub>PLAZ</sub>	1	PSEN low to address float		10		10	ns
<b>Data Memory</b>							
t <sub>RLRH</sub>	2, 3	RD pulse width	275		6t <sub>CLCL</sub> -100		ns
t <sub>WLWH</sub>	2, 3	WR pulse width	275		6t <sub>CLCL</sub> -100		ns
t <sub>RLDV</sub>	2, 3	RD low to valid data in		147		5t <sub>CLCL</sub> -165	ns
t <sub>RHDZ</sub>	2, 3	Data hold after RD	0		0		ns
t <sub>RHDZ</sub>	2, 3	Data float after RD		65		2t <sub>CLCL</sub> -60	ns
t <sub>LLDV</sub>	2, 3	ALE low to valid data in		350		8t <sub>CLCL</sub> -150	ns
t <sub>AVDV</sub>	2, 3	Address to valid data in		397		9t <sub>CLCL</sub> -165	ns
t <sub>LLWL</sub>	2, 3	ALE low to RD or WR low	137	237	3t <sub>CLCL</sub> -50	3t <sub>CLCL</sub> +50	ns
t <sub>AVWL</sub>	2, 3	Address valid to WR low or RD low	175		4t <sub>CLCL</sub> -75		ns
t <sub>QVWX</sub>	2, 3	Data valid to WR transition	42		t <sub>CLCL</sub> -20		ns
t <sub>WHQX</sub>	2, 3	Data hold after WR	42		t <sub>CLCL</sub> -20		ns
t <sub>QVWH</sub>	3	Data valid to WR high	287		7t <sub>CLCL</sub> -150		ns
t <sub>RLAZ</sub>	2, 3	RD low to address float		0		0	ns
t <sub>WLHL</sub>	2, 3	RD or WR high to ALE high	40	87	t <sub>CLCL</sub> -20	t <sub>CLCL</sub> +25	ns
<b>External Clock</b>							
t <sub>CHCX</sub>	5	High time	12		20		ns
t <sub>CLCX</sub>	5	Low time	12		20		ns
t <sub>CLCH</sub>	5	Rise time		20		20	ns
t <sub>CHCL</sub>	5	Fall time		20		20	ns
<b>Shift Register</b>							
t <sub>XLXL</sub>	4	Serial port clock cycle time	1		12t <sub>CLCL</sub>		μs
t <sub>QVXH</sub>	4	Output data setup to clock rising edge	492		10t <sub>CLCL</sub> -133		ns
t <sub>WHQX</sub>	4	Output data hold after clock rising edge	8		2t <sub>CLCL</sub> -117		ns
t <sub>XHDZ</sub>	4	Input data hold after clock rising edge	0		0		ns
t <sub>XHDV</sub>	4	Clock rising edge to input data valid		492		10t <sub>CLCL</sub> -133	ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- Interfacing the 87C51FB to devices with float times up to 45ns is permitted. This limited bus contention will not cause damage to Port 0 drivers.

## CMOS single-chip 8-bit microcontroller 83C504/87C504

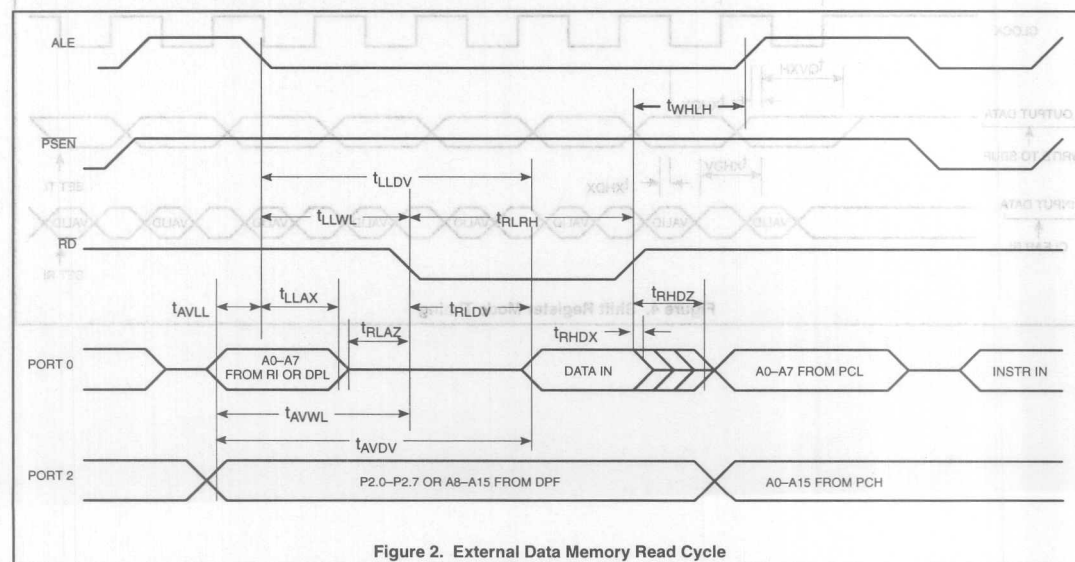
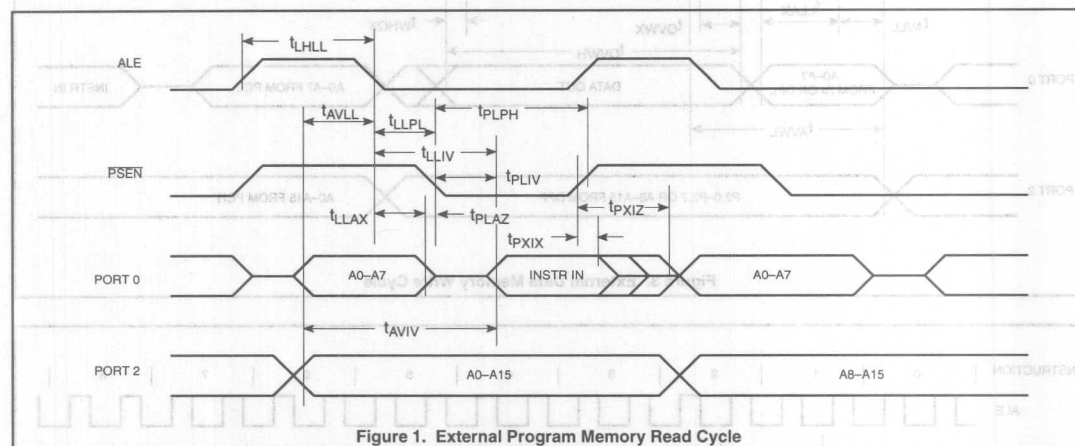
## EXPLANATION OF THE AC SYMBOLS

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

A – Address  
C – Clock  
D – Input data  
H – Logic level high  
I – Instruction (program memory contents)  
L – Logic level low, or ALE

P – PSEN  
Q – Output data  
R – RD signal  
t – Time  
V – Valid  
W – WR signal  
X – No longer a valid logic level  
Z – Float

Examples:  $t_{AVLL}$  = Time for address valid to ALE low.  
 $t_{LLPL}$  = Time for ALE low to PSEN low.



## CMOS single-chip 8-bit microcontroller 83C504/87C504

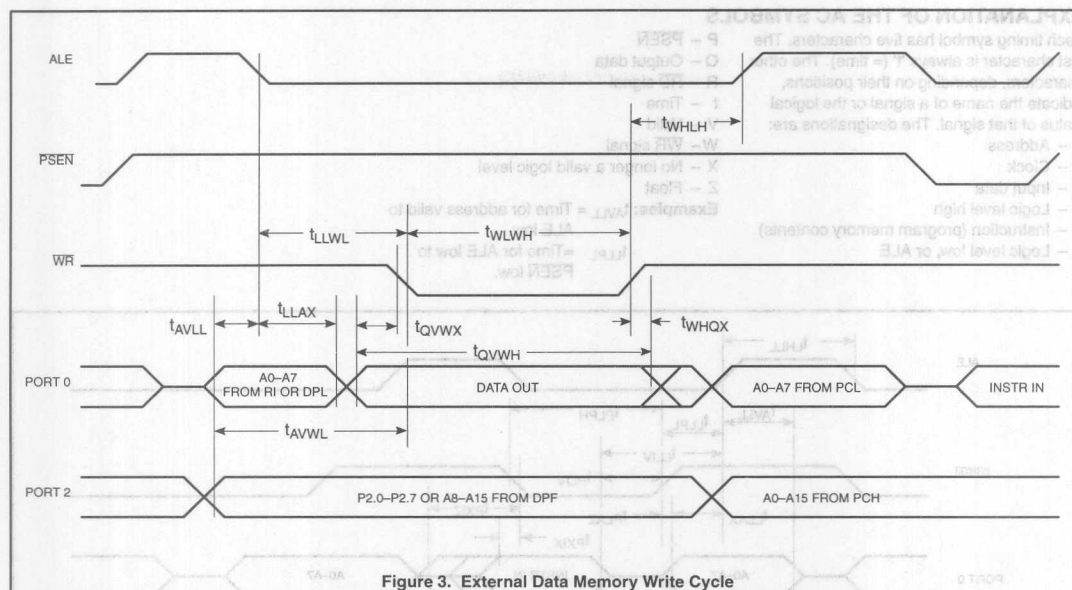


Figure 3. External Data Memory Write Cycle

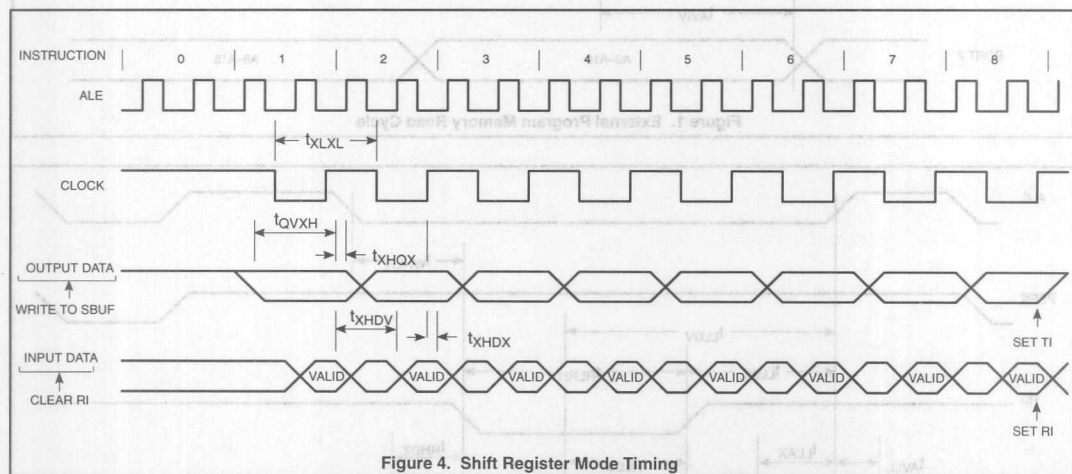
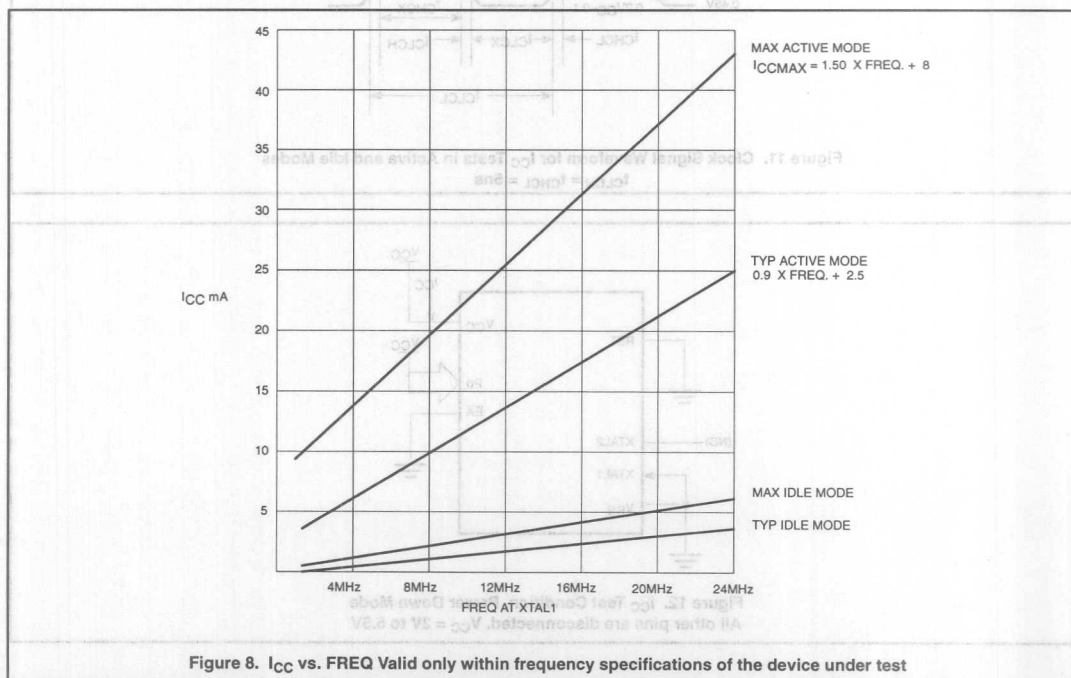
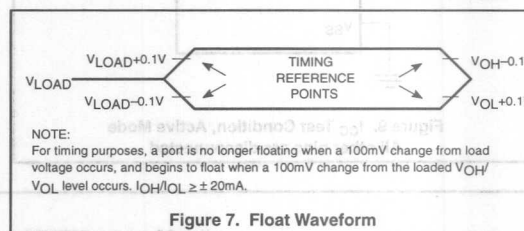
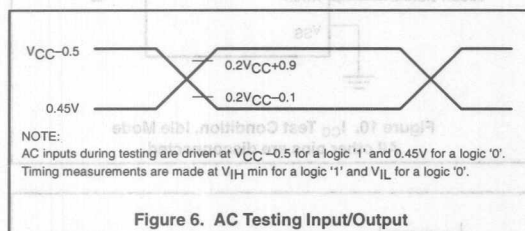
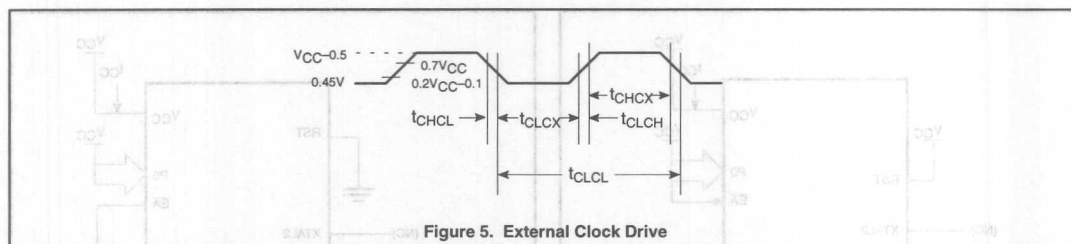


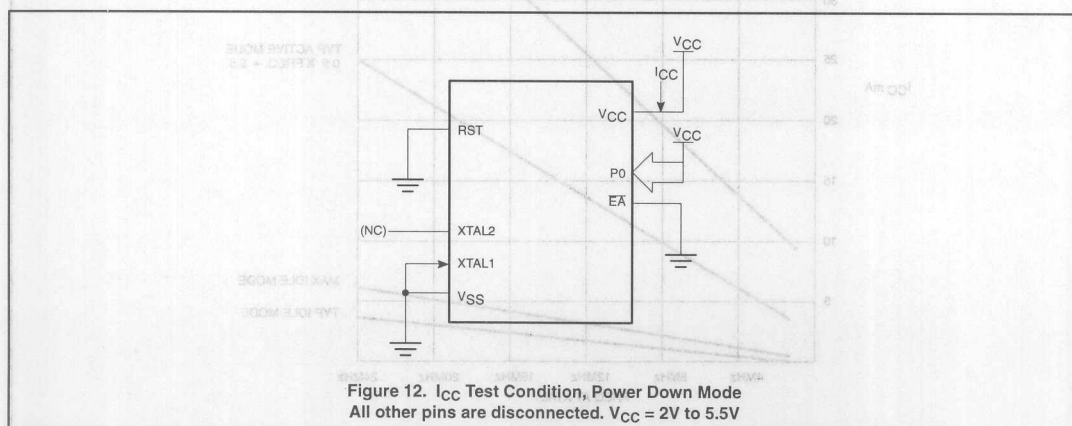
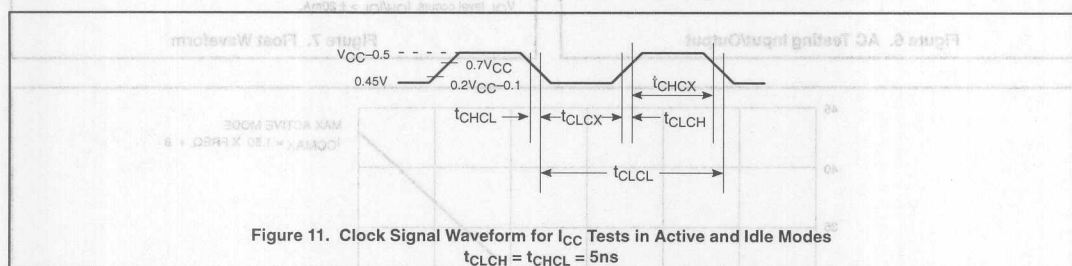
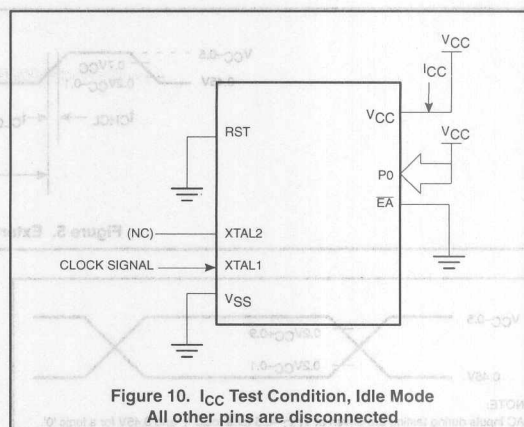
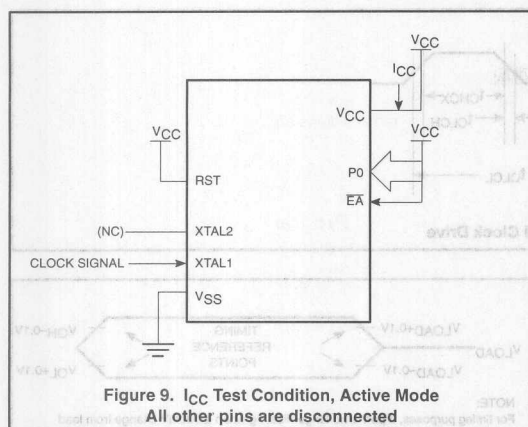
Figure 4. Shift Register Mode Timing



## CMOS single-chip 8-bit microcontroller 83C504/87C504



## CMOS single-chip 8-bit microcontroller 83C504/87C504



## CMOS single-chip 8-bit microcontroller

83C504/87C504

**EPROM CHARACTERISTICS**

The 87C504 is programmed by using a modified Quick-Pulse Programming™ algorithm. It differs from older methods in the value used for  $V_{PP}$  (programming supply voltage) and in the width and number of the ALE/PROG pulses.

The 87C504 contains two signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes identify the device as an 87C504 manufactured by Philips.

Table 3 shows the logic levels for reading the signature byte, and for programming the program memory, the encryption table, and the security bits. The circuit configuration and waveforms for quick-pulse programming are shown in Figures 13 and 14. Figure 15 shows the circuit configuration for normal program memory verification.

**Quick-Pulse Programming**

The setup for microcontroller quick-pulse programming is shown in Figure 13. Note that the 87C504 is running with a 4 to 6MHz oscillator. The reason the oscillator needs to be running is that the device is executing internal address and program data transfers.

The address of the EPROM location to be programmed is applied to ports 1 and 2, as shown in Figure 13. The code byte to be programmed into that location is applied to port 0. RST, PSEN and pins of ports 2 and 3 specified in Table 3 are held at the 'Program Code Data' levels indicated in Table 3. The ALE/PROG is pulsed low 25 times as shown in Figure 14.

To program the encryption table, repeat the 25 pulse programming sequence for addresses 0 through 1FH, using the 'Pgm Encryption Table' levels. Do not forget that after the encryption table is programmed, verification cycles will produce only encrypted data.

To program the security bits, repeat the 25 pulse programming sequence using the 'Pgm Security Bit' levels. After one security bit is programmed, further programming of the code memory and encryption table is disabled. However, the other security bit can still be programmed.

Note that the  $\overline{EA}/V_{PP}$  pin must not be allowed to go above the maximum specified  $V_{PP}$  level for any amount of time. Even a narrow glitch above that voltage can cause permanent damage to the device. The  $V_{PP}$  source should be well regulated and free of glitches and overshoot.

**Program Verification**

If security bit 2 has not been programmed, the on-chip program memory can be read out for program verification. The address of the program memory locations to be read is applied to ports 1 and 2 as shown in Figure 15. The other pins are held at the 'Verify Code Data' levels indicated in Table 3. The contents of the address location will be emitted on port 0. External pull-ups are required on port 0 for this operation.

If the encryption table has been programmed, the data presented at port 0 will be the exclusive NOR of the program byte with one of the encryption bytes. The user will have to know the encryption table contents in order to correctly decode the verification data. The encryption table itself cannot be read out.

**Reading the Signature Bytes**

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. The values are:

(030H) = 15H indicates manufactured by Philips

(031H) = BBH indicates 87C504

**Program/Verify Algorithms**

Any algorithm in agreement with the conditions listed in Table 3, and which satisfies the timing specifications, is suitable.

**Erase Characteristics**

Erase of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent erasure. **For this and secondary effects, it is recommended that an opaque label be placed over the window.** For elevated temperature or environments where solvents are being used, apply Kapton tape Fluorglas part number 2345-5, or equivalent.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 angstroms) to an integrated dose of at least 15W-s/cm<sup>2</sup>. Exposing the EPROM to an ultraviolet lamp of 12,000µW/cm<sup>2</sup> rating for 20 to 39 minutes, at a distance of about 1 inch, should be sufficient.

Erase leaves the array in an all 1s state.

**Table 3. EPROM Programming Modes**

MODE	RST	PSEN	ALE/PROG	$\overline{EA}/V_{PP}$	P2.7	P2.6	P3.7	P3.6
Read signature	1	0	1	1	0	0	0	0
Program code data	1	0	0*	$V_{PP}$	1	0	1	1
Verify code data	1	0	1	1	0	0	1	1
Pgm encryption table	1	0	0*	$V_{PP}$	1	0	1	0
Pgm security bit 1	1	0	0*	$V_{PP}$	1	1	1	1
Pgm security bit 2	1	0	0*	$V_{PP}$	1	1	0	0

**NOTES:**

1. '0' = Valid low for that pin, '1' = valid high for that pin.

2.  $V_{PP}$  = 12.75V  $\pm$  0.25V.

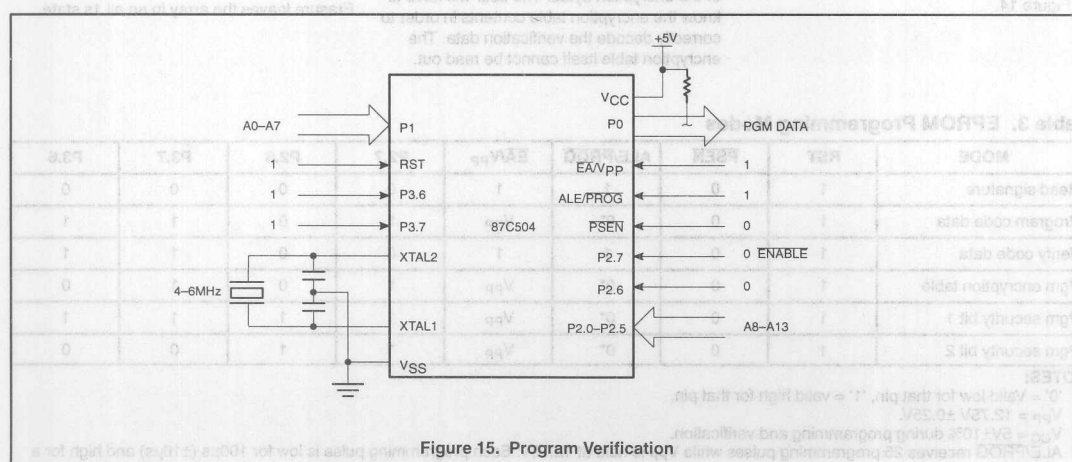
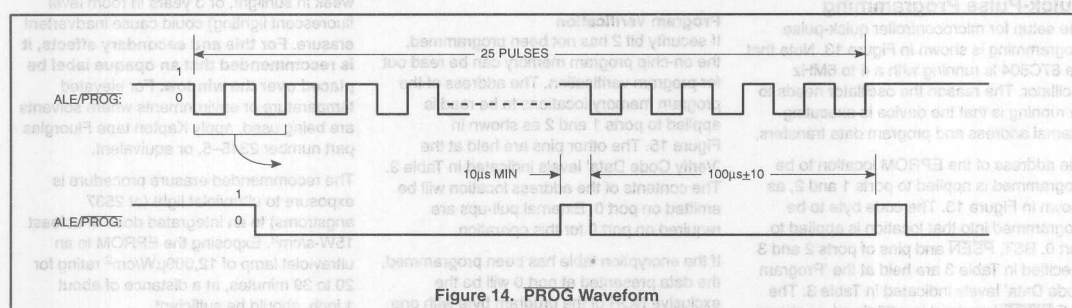
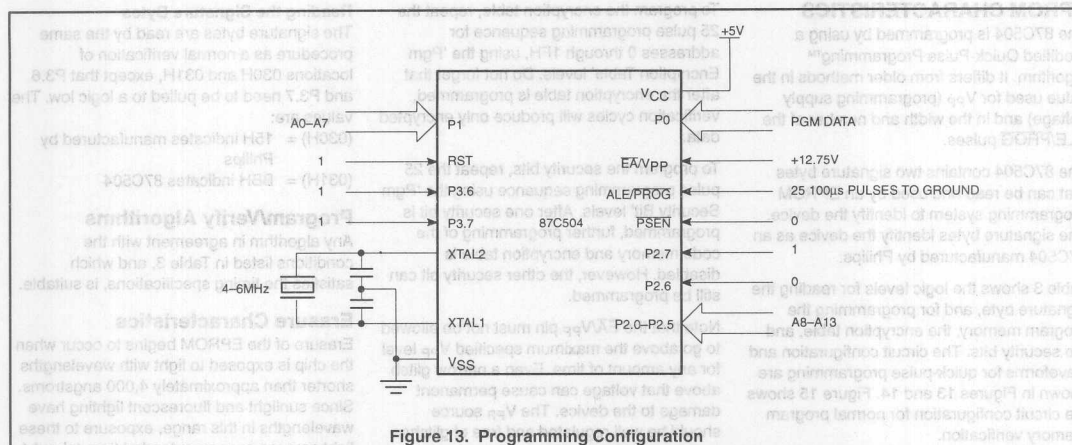
3.  $V_{CC}$  = 5V  $\pm$  10% during programming and verification.

\* ALE/PROG receives 25 programming pulses while  $V_{PP}$  is held at 12.75V. Each programming pulse is low for 100µs ( $\pm$  10µs) and high for a minimum of 10µs.

™Trademark phrase of Intel Corporation.

## CMOS single-chip 8-bit microcontroller

87C504/87C504



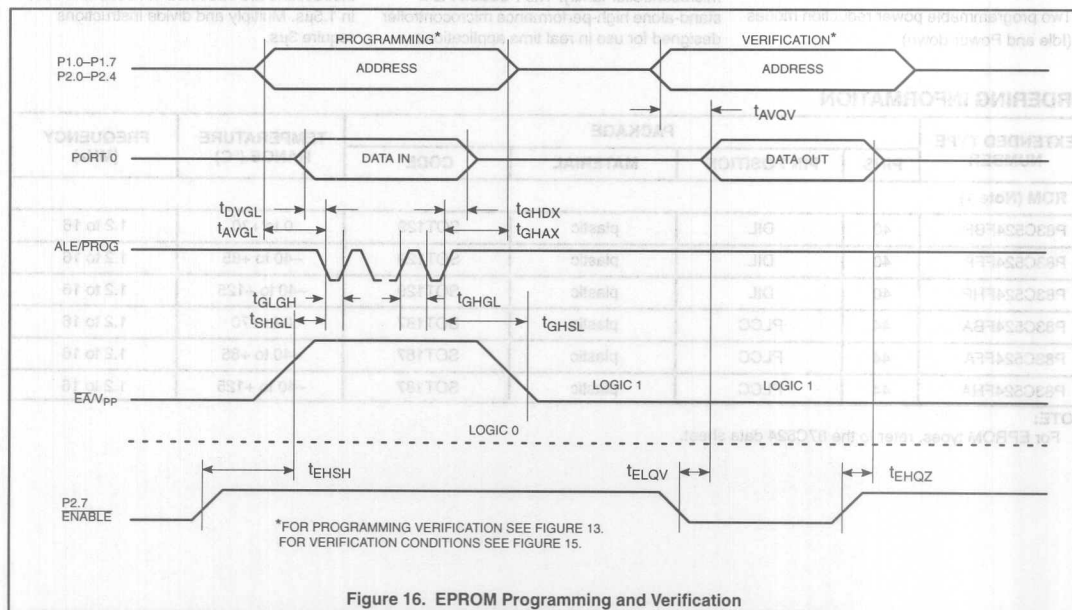
## CMOS single-chip 8-bit microcontroller

83C504/87C504

## EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

T<sub>amb</sub> = 21°C to +27°C, V<sub>CC</sub> = 5V±10%, V<sub>SS</sub> = 0V (See Figure 16)

SYMBOL	PARAMETER	MIN	MAX	UNIT
V <sub>PP</sub>	Programming supply voltage	12.5	13.0	V
I <sub>PP</sub>	Programming supply current		50	mA
1/t <sub>CLCL</sub>	Oscillator frequency	4	6	MHz
t <sub>AVGL</sub>	Address setup to PROG low	48t <sub>CLCL</sub>		
t <sub>GHAX</sub>	Address hold after PROG	48t <sub>CLCL</sub>		
t <sub>DVGL</sub>	Data setup to PROG low	48t <sub>CLCL</sub>		
t <sub>GHDX</sub>	Data hold after PROG	48t <sub>CLCL</sub>		
t <sub>EHS</sub>	P2.7 (ENABLE) high to V <sub>PP</sub>	48t <sub>CLCL</sub>		
t <sub>SHGL</sub>	V <sub>PP</sub> setup to PROG low	10		μs
t <sub>GHSL</sub>	V <sub>PP</sub> hold after PROG	10		μs
t <sub>GLGH</sub>	PROG width	90	110	μs
t <sub>AVQV</sub>	Address to data valid		48t <sub>CLCL</sub>	
t <sub>ELQV</sub>	ENABLE low to data valid		48t <sub>CLCL</sub>	
t <sub>EHQV</sub>	Data float after ENABLE	0	48t <sub>CLCL</sub>	
t <sub>GHGL</sub>	PROG high to PROG low	10		μs





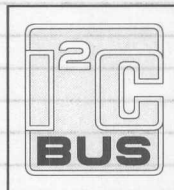
## 8-bit microcontroller

## P83C524

## FEATURES

- 80C51 CPU
- 16 kbytes on-chip ROM, expandable externally to 64 kbytes Program Memory address space
- 512 bytes on-chip RAM, expandable externally to 64 kbytes Data Memory address space
- Four 8-bit I/O ports
- Full-duplex UART compatible with the standard 80C51 and the 8052
- Two standard 16-bit timer/counters
- An additional 16-bit timer (functionally equivalent to the timer 2 of the 8052)
- On-chip Watchdog Timer (WDT) with a separate on-chip oscillator
- Bit-level I<sup>2</sup>C-bus hardware serial I/O Port
- 7-source and 7-vector interrupt structure with 2 priority levels
- Up to 3 external interrupt request inputs
- Two programmable power reduction modes (Idle and Power-down)

- Termination of Idle mode by any interrupt, external or WDT (watchdog) reset
- Wake-up from Power-down by external interrupt, external or WDT reset
- ROM code protection
- XTAL frequency range: 1.2MHz to 16MHz
- All packaging pin-outs fully compatible to the standard 8051/8052.



## GENERAL DESCRIPTION

The P83C524 8-bit microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The P83C524 is a stand-alone high-performance microcontroller designed for use in real time applications

such as instrumentation, industrial control, medium to high-end consumer applications and specific automotive control applications.

The P83C524 contains a non-volatile 16K × 8 read-only program memory, a volatile 512 × 8 read/write data memory, four 8-bit I/O ports, two 16-bit timer/event counters (identical to the timers of the 80C51), a 16-bit timer (identical to the timer 2 of the 8052), a multi-source, two-priority-level, nested interrupt structure, two serial interfaces (UART and bit-level I<sup>2</sup>C-bus), an on-chip oscillator and timing circuits, a watchdog timer (WDT) with a separate on-chip oscillator. For systems that require extra capability, the P83C524 can be expanded using standard TTL compatible memories and logic.

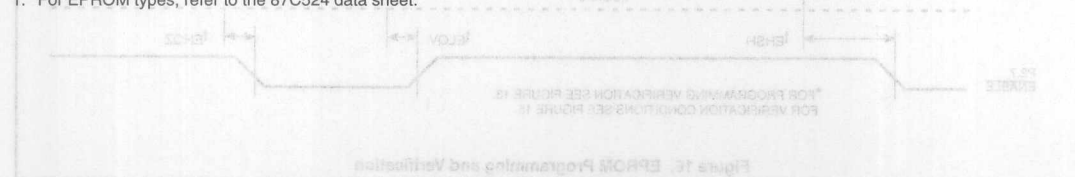
The device also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The P83C524 has the same instruction set as the PCB80C51 which consists of over 100 instructions: 49 one-byte, 46 two-byte and 16 three-byte. With a 16MHz crystal, 58% of the instructions are executed in 750ns and 40% in 1.5µs. Multiply and divide instructions require 3µs.

## ORDERING INFORMATION

EXTENDED TYPE NUMBER	PACKAGE				TEMPERATURE RANGE (°C)	FREQUENCY (MHz)
	PINS	PIN POSITION	MATERIAL	CODE		
ROM (Note 1)						
P83C524FBP	40	DIL	plastic	SOT129	0 to +70	1.2 to 16
P83C524FFP	40	DIL	plastic	SOT129	-40 to +85	1.2 to 16
P83C524FHP	40	DIL	plastic	SOT129	-40 to +125	1.2 to 16
P83C524FBA	44	PLCC	plastic	SOT187	0 to +70	1.2 to 16
P83C524FFA	44	PLCC	plastic	SOT187	-40 to +85	1.2 to 16
P83C524FHA	44	PLCC	plastic	SOT187	-40 to +125	1.2 to 16

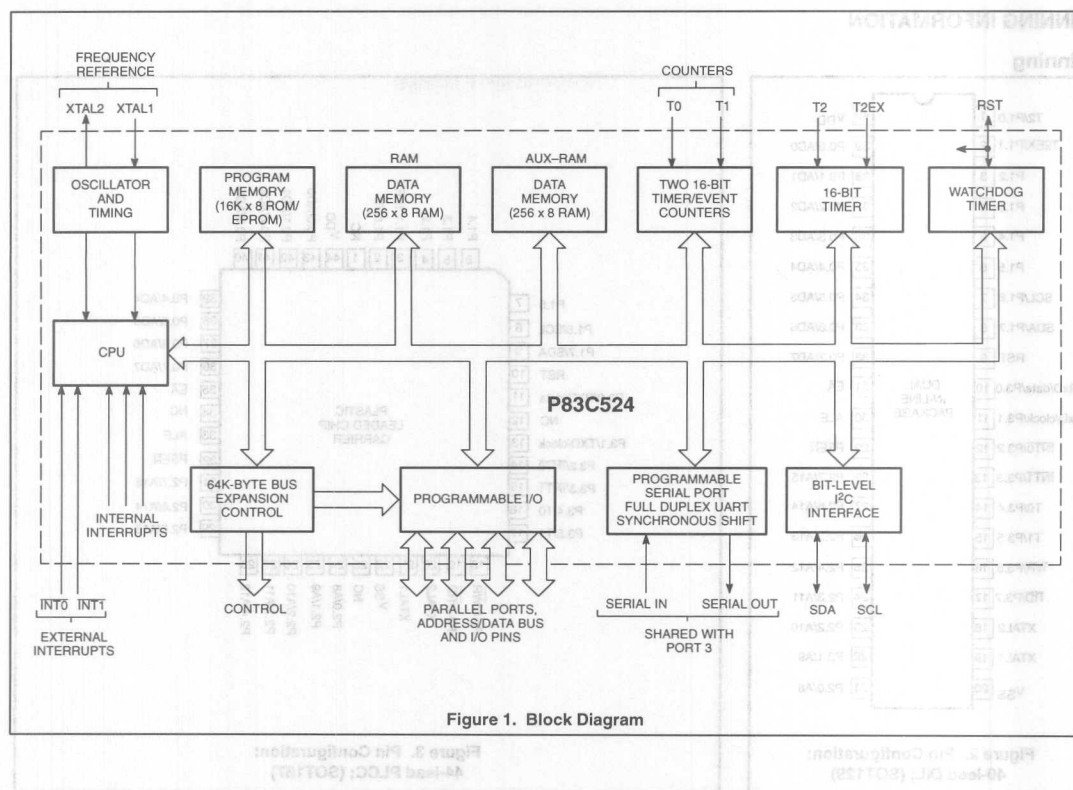
## NOTE:

1. For EPROM types, refer to the 87C524 data sheet.



## 8-bit microcontroller

P83C524

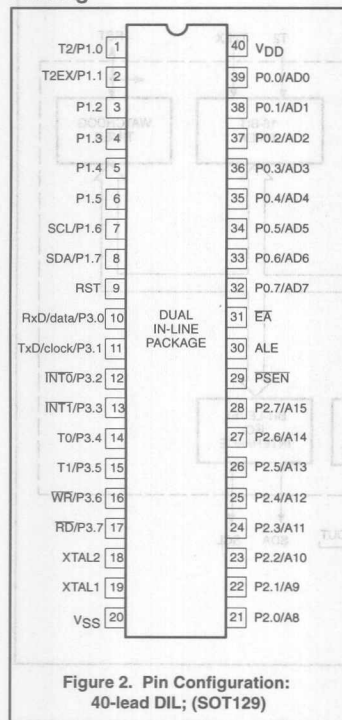
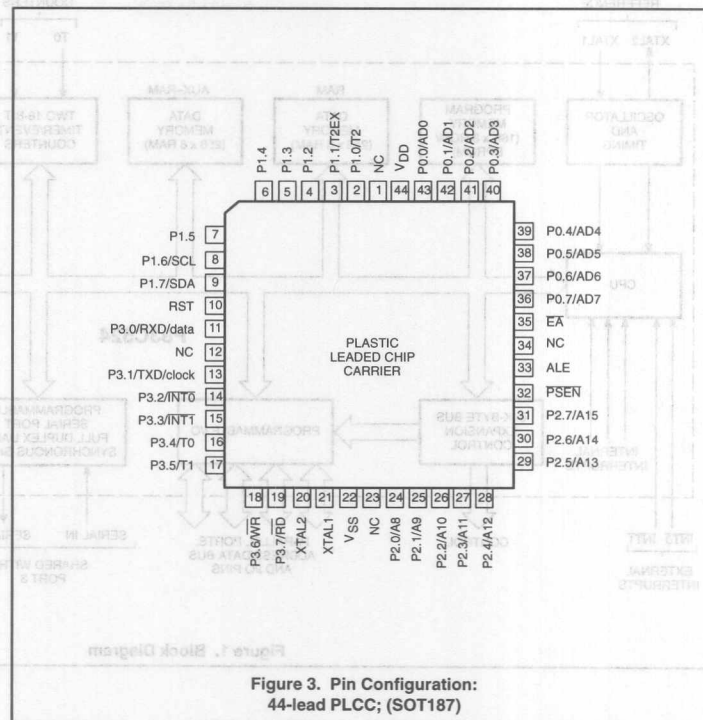


## 8-bit microcontroller

P83C524

## PINNING INFORMATION

## Pinning

Figure 2. Pin Configuration:  
40-lead DIL; (SOT129)Figure 3. Pin Configuration:  
44-lead PLCC; (SOT187)

## 8-bit microcontroller

P83C524

## PIN DESCRIPTION

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	PLCC		
P1.0–P1.7	1–8	2–9 (1 NC)	I/O	<p><b>Port 1:</b> 8-bit quasi-bidirectional I/O port. Port 1 can sink/source one TTL (4 LS TTL) input. It can drive CMOS inputs without external pull-ups, except P1.6 and P1.7 which have open drain outputs.</p> <p><b>Port 1 alternative functions:</b></p> <p><b>T2 (P1.0):</b> Timer/event counter 2 external event counter input (falling edge triggered).</p> <p><b>T2EX (P1.1):</b> Timer/event counter 2 capture/reload trigger or external interrupt 2 input (falling edge triggered)</p> <p><b>SCL (P1.6):</b> I<sup>2</sup>C serial port clock line.</p> <p><b>SDA (P1.7):</b> I<sup>2</sup>C serial port data line.</p>
RST	9	10	I/O	<p><b>Reset:</b> A HIGH level on this pin for two machine cycles while the oscillator is running, resets the device. An internal pull-down resistor permits power-on reset using only a capacitor connected to V<sub>DD</sub>. After a WDT overflow, this pin is pulled HIGH while the internal reset signal is active.</p>
P3.0–P3.7	10–17	11, 13–19 (12 NC)	I/O	<p><b>Port 3:</b> 8-bit quasi-bidirectional I/O port with internal pull-ups. Port 3 can sink/source one TTL (=4 LS TTL) input. It can drive CMOS inputs without external pull-ups.</p> <p><b>Port 3 alternative functions:</b></p> <p><b>RxD/data (P3.0):</b> Serial Port data input (asynchronous) or data input/output (synchronous).</p> <p><b>TxD/clock (P3.1):</b> Serial Port data output (asynchronous) or clock output (synchronous).</p> <p><b>INT0 (P3.2):</b> External interrupt 0 or gate control input for timer/event counter 0.</p> <p><b>INT1 (P3.3):</b> External interrupt 1 or gate control input for timer/event counter 1.</p> <p><b>T0 (P3.4):</b> External input for timer/event counter 0.</p> <p><b>T1 (P3.5):</b> External input for timer/event counter 1.</p> <p><b>WR (P3.6):</b> External data memory write strobe.</p> <p><b>RD (P3.7):</b> External data memory read strobe.</p>
XTAL2	18	20	O	<p><b>Crystal input 2:</b> Output of the inverting amplifier that forms the oscillator. This pin left open-circuit when an external oscillator clock is used.</p>
XTAL1	19	21	I	<p><b>Crystal input 1:</b> Input to the inverting amplifier that forms the oscillator, and input to the internal clock generator. Receives the external oscillator clock signal when an external oscillator is used.</p>
V <sub>SS</sub>	20	22	I	<p><b>Ground:</b> Circuit ground potential.</p>
P2.0–P2.7 (A8 to A15)	21–28	24–31 (23 NC)	I/O	<p><b>Port 2:</b> 8-bit quasi-bidirectional I/O port with internal pull-ups. During access to external memories (RAM/ROM) that use 16-bit addresses (MOVX @DPTR) Port 2 emits the high-order address byte (A8 to A15). Port 2 can sink/source one TTL (=4 LS TTL) input. It can drive CMOS inputs without external pull-ups.</p>
PSEN	29	32	O	<p><b>Program Store Enable output:</b> Read strobe to external program memory via Port 0 and Port 2. It is activated twice each machine cycle during fetches from external program memory. When executing out of external program memory two activations of PSEN are skipped during each access to external data memory. PSEN is not activated (remains HIGH) during fetches from external program memory. PSEN can sink source 8 LS TTL inputs. It can drive CMOS inputs without external pull-ups.</p>
ALE	30	33	I/O	<p><b>Address Latch Enable output:</b> Latches the LOW byte of the address during access to external memory in normal operation. It is activated every six oscillator periods except during an external data memory access. ALE can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pull-up.</p>
E <sub>A</sub>	31	35 (34 NC)	I	<p><b>External Access input:</b> When during RESET, E<sub>A</sub> is held at a TTL HIGH level, the CPU executes out of the internal program ROM, provided the program counter is less than 32768. When E<sub>A</sub> is held at a TTL LOW level during RESET, the CPU executes out of external program memory via Port 0 and Port 2. E<sub>A</sub> is not allowed to float.</p>
P0.0–0.7 (AD0 to AD7)	32–39	36–43	I/O	<p><b>Port 0:</b> 8-bit open drain bidirectional I/O Port. It is also the multiplexed low-order address and data bus during accesses to external memory (AD0 to AD7). During these accesses internal pull-ups are activated. Port 0 can sink/source 8 LS TTL inputs.</p>
V <sub>DD</sub>	40	44	I	<p><b>Power Supply:</b> +5V power supply pin during normal operation, Idle mode and Power-down mode.</p> <p>To avoid a "latch-up" effect at power-on, the voltage on any pin (at any time) must not be higher than V<sub>DD</sub> +0.5V or lower than V<sub>SS</sub> –0.5V respectively.</p>
NC	–	1, 12, 23, 34	–	<p>No connection (PLCC only).</p>

## 8-bit microcontroller

P83C524

Table 1. 8XC524/8XC528 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB							LSB	
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
DPTR: DPH DPL	Data pointer (2 bytes): Data pointer high Data pointer low	83H 82H									00H 00H
IE*#	Interrupt enable	A8H	AF	AE	AD	AC	AB	AA	A9	A8	00H
			EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0	00H
IP*#	Interrupt priority	B8H	—	PS1	PT2	PS0	PT1	PX1	PT0	PX0	x0000000B
P0*	Port 0	80H	87	86	85	84	83	82	81	80	FFH
			AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
P1*	Port 1	90H	97	96	95	94	93	92	91	90	FFH
			SDA	SEL	—	—	—	—	T2EX	T2	FFH
P2*	Port 2	A0H	A7	A6	A5	A4	A3	A2	A1	A0	FFH
			A15	A14	A13	A12	A11	A10	A9	A8	FFH
P3*	Port 3	B0H	B7	B6	B5	B4	B3	B2	B1	B0	FFH
			RD	WR	T1	T0	INT1	INT0	TxD	RxD	FFH
PCON	Power control	87H	SMOD	—	—	—	GF1	GF0	PD	IDL	0xxx0000B
			D7	D6	D5	D4	D3	D2	D1	D0	00H
PSW*	Program status word	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	00H
RCAP2H#	Capture high	CBH									00H
RCAP2L#	Capture low	CAH									00H
SBUF	Serial data buffer	99H	9F	9E	9D	9C	9B	9A	99	98	xxxxxxx0B
SCON*	Serial controller	98H	SM0	SM1	SM2	REN	TB8	RB8	T1	RI	00H
S1BIT#	Serial I <sup>2</sup> C data	D9H/RD	SD1	0	0	0	0	0	0	0	x0000000B
		WR	SD0	X	X	X	X	X	X	X	0xxxxxxx0B
S1INT#	Serial I <sup>2</sup> C interrupt	DAH	INT	X	X	X	X	X	X	X	0xxxxxxx0B
			DF	DE	DD	DC	DB	DA	D9	D8	00H
S1SCS*#	Serial I <sup>2</sup> C control	D8H/RD	SD1	SCI	CLH	BB	RBF	WBF	STR	ENS	xxxx0000B
		WR	SD0	SC0	CLH	X	X	X	STR	ENS	00xxxx00B
SP	Stack pointer	81H	8F	8E	8D	8C	8B	8A	89	88	07H
TCON*	Timer control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
			CF	CE	CD	CC	CB	CA	C9	C8	00H
T2CON*#	Timer 2 control	C8H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	00H
TH0	Timer high 0	8CH									00H
TH1	Timer high 1	8DH									00H
TH2#	Timer high 2	CDH									00H
TL0	Timer low 0	8AH									00H
TL1	Timer low 1	8BH									00H
TL2#	Timer low 2	CCH									00H
T3#	Watchdog timer	FFH									00H
TMOD	Timer mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H
WDCON#	Watchdog control	A5H									A5H

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.



## 8-bit microcontroller

## P83C524

**WATCHDOG TIMER T3**

The watchdog timer (WDT) (see Figure 4) consists of an 11-bit prescaler and an 8-bit timer formed by special function register T3. The prescaler is incremented by an on-chip oscillator with a fixed frequency of 1MHz. The maximum tolerance on this frequency is -50% and +100%. The 8-bit timer increments every 2048 cycles of the on-chip oscillator. When a timer overflow occurs, the microcontroller is reset and a reset output pulse of  $16 \times 2048$  cycles of the on-chip oscillator is generated at pin RST. The internal RESET signal is not inhibited when the external RST pin is kept LOW by, for example, an external reset circuit. The RESET signal drives Ports 1, 2 and 3 outputs into the HIGH state and port 0 into high impedance, whether the XTAL-clock is running or not.

The watchdog timer is controlled by special function register WDCON with the direct address location A5H. WDCON can be read and written by software. A value of A5H in WDCON halts the on-chip oscillator and clears both the prescaler and timer T3. After

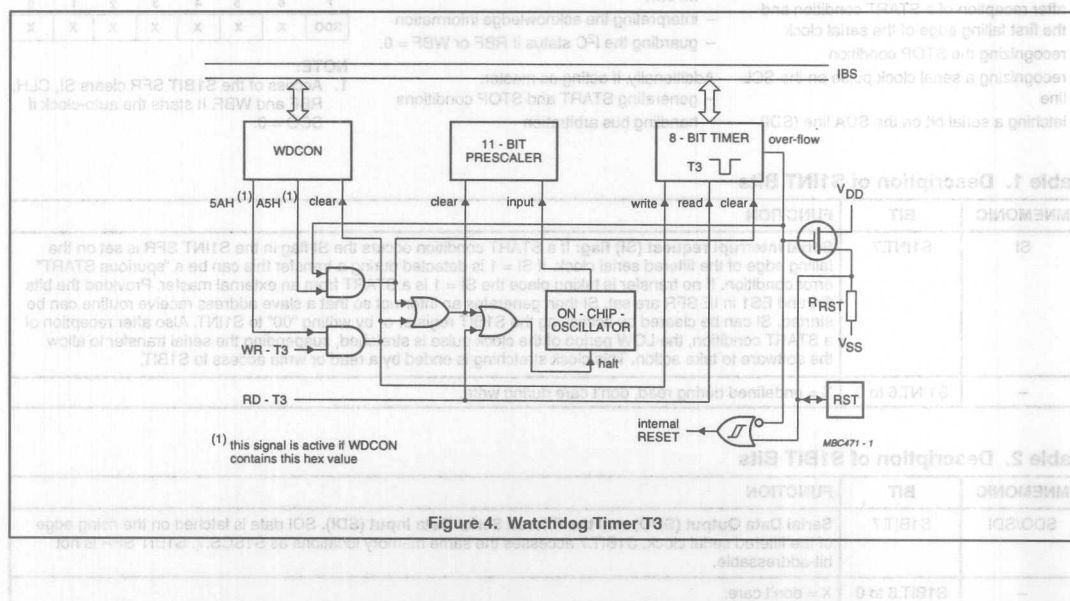
the RESET signal, WDCON contains A5H. Every value other than A5H in WDCON enables the watchdog timer. When the watchdog timer is enabled, it runs independently of the XTAL-clock.

Timer T3 can be read on the fly. Timer T3 can be written to only if WDCON has previously been loaded with 5AH, otherwise T# and the prescaler are not affected. A successful write operation to T3 also clears the prescaler and WDCON. During a read or write operation addressing T3, the output of the on-chip oscillator is inhibited to prevent timing problems due to asynchronous increments of T3. To prevent an overflow of the watchdog timer, the user program has to reload T3 within periods that are shorter than the programmed watchdog timer interval. This time interval is determined by the 8-bit reload value that is written into register T3.

$$\text{Watchdog timer interval} = \frac{[256 - (T3)] \times 2048}{\text{on-chip oscillator frequency}}$$

The advantages of this implementation are:

- Only an internal reset connection to the microcontroller core.
- The Power-down mode and the Watchdog (WDT) function can be used concurrently.
- The WDT also monitors the XTAL oscillator. In case of a failure the port outputs are forced to a defined HIGH state.
- Interference will not disable the WDT because it is unlikely that it will force WDCON to A5H.
- Tolerances of the on-chip oscillator can be adjusted by testing the T3 value and adapting the reload value.
- The WDT can be enabled and disabled under control of the user software. This gives the possibility to use both the Watchdog function and the Power-down function.
- The direct address A5H of WDCON and its disable value A5H will not unintentionally be present at a random location in the field of program code, except for immediate data, because the opcode A5H is not used in the instruction set.



## 8-bit microcontroller

P83C524

**BIT-LEVEL I<sup>2</sup>C INTERFACE**

This bit-level serial I/O interface supports the I<sup>2</sup>C-bus (see Figure 5). P1.6/SCL and P1.7/SDA are the serial I/O pins. These two pins meet the I<sup>2</sup>C specification concerning the input levels and output drive capability. Consequently, these pins have an open drain output configuration. All the four modes of the I<sup>2</sup>C-bus are supported:

- master transmitter
- master receiver
- slave transmitter
- slave receiver

The advantages of the bit-level I<sup>2</sup>C hardware compared with a full software I<sup>2</sup>C implementation are:

- The hardware can generate the SCL pulse
- Testing a single bit (RBF respectively, WBF) is sufficient as a check for error free transmission.

The bit-level I<sup>2</sup>C hardware operates on serial bit level and performs the following functions:

- filtering the incoming serial data and clock signals
- recognizing the START condition
- generating a serial interrupt request SI after reception of a START condition and the first falling edge of the serial clock
- recognizing the STOP condition
- recognizing a serial clock pulse on the SCL line
- latching a serial bit on the SDA line (SDI)

- stretching the SCL LOW period of the serial clock to suspend the transfer of the next serial data bit
- setting Read Bit Finished (RBF) when the SCL clock pulse has finished and Write Bit Finished (WBF) if there is no arbitration loss detected (i.e., SDA = 0 while SDO = 1)
- setting a serial clock Low-to-High detected (CLH) flag
- setting a Bus Busy (BB) flag on a START condition and clearing this flag on a STOP condition
- releasing the SCL line and clearing the CLH, RBF and WBF flags to resume transfer of the next serial data bit
- generating an automatic clock if the single bit data register S1BIT is used in master mode.

The following functions must be done in software:

- handling the I<sup>2</sup>C START interrupts
  - converting serial to parallel data when receiving
  - converting parallel to serial data when transmitting
  - comparing the received slave address with its own
  - interpreting the acknowledge information
  - guarding the I<sup>2</sup>C status if RBF or WBF = 0.
- Additionally, if acting as master:
- generating START and STOP conditions
  - handling bus arbitration

- generating serial clock pulses if S1BIT is not used.

Three SFRs control the bit-level I<sup>2</sup>C interface: S1INT, S1BIT and S1SCS.

**S1INT: I<sup>2</sup>C Interrupt Register**

This register is located at address DAH. Refer to Table 1.

**S1INT SFR (DAH)**

	7	6	5	4	3	2	1	0
SI	X	X	X	X	X	X	X	X

**NOTE:**

1. **SI bit:** Writing a logic 0 clears this bit, writing a logic 1 has no effect.

**S1BIT: Single-bit Data Register with I<sup>2</sup>C Auto-Clock**

This register is located at address D9H. Refer to Table 2.

**S1BIT SFR (D9H)****READ**

	7	6	5	4	3	2	1	0
SDI	0	0	0	0	0	0	0	0

**WRITE**

	7	6	5	4	3	2	1	0
SDO	X	X	X	X	X	X	X	X

**NOTE:**

1. Access of the S1BIT SFR clears SI, CLH, RBF and WBF. It starts the auto-clock if SCO = 0.

**Table 1. Description of S1INT Bits**

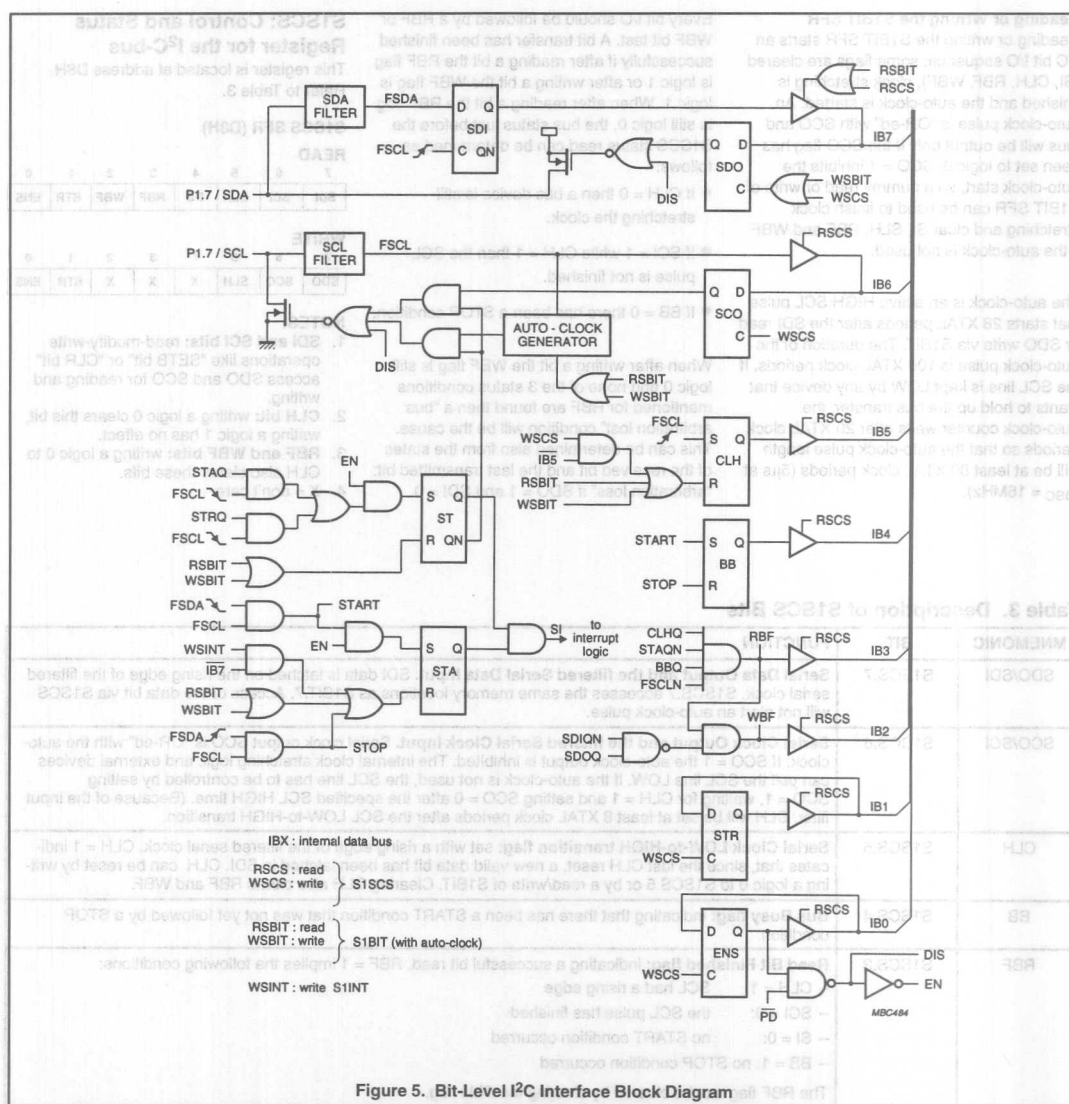
MNEMONIC	BIT	FUNCTION
SI	S1INT.7	<b>Serial Interrupt request (SI) flag:</b> If a START condition occurs the SI flag in the S1INT SFR is set on the falling edge of the filtered serial clock. If SI = 1 is detected during a transfer this can be a "spurious START" error condition. If no transfer is taking place the SI = 1 is a START from an external master. Provided the bits EA and ES1 in IE SFR are set, SI then generates an interrupt so that a slave address receive routine can be started. SI can be cleared by accessing the S1BIT register or by writing "00" to S1INT. Also after reception of a START condition, the LOW period of the clock pulse is stretched, suspending the serial transfer to allow the software to take action. This clock stretching is ended by a read or write access to S1BIT.
–	S1INT.6 to 0	X = undefined during read, don't care during write.

**Table 2. Description of S1BIT Bits**

MNEMONIC	BIT	FUNCTION
SDO/SDI	S1BIT.7	<b>Serial Data Output (SDO) and the filtered Serial Data Input (SDI).</b> SDI data is latched on the rising edge of the filtered serial clock. S1BIT.7 accesses the same memory locations as S1SCS.7. S1BIT SFR is not bit-addressable.
–	S1BIT.6 to 0	X = don't care.

## 8-bit microcontroller

## P83C524



## 8-bit microcontroller

P83C524

**Reading or Writing the S1BIT SFR**

Reading or writing the S1BIT SFR starts an I<sup>2</sup>C bit I/O sequence: some flags are cleared (SI, CLH, RBF, WBF), clock stretching is finished and the auto-clock is started. An auto-clock pulse is "OR-ed" with SCO and thus will be output only if the SCO flag has been set to logic 0. SCO = 1 inhibits the auto-clock start, so a dummy read or write of S1BIT SFR can be used to finish clock stretching and clear SI, SLH, RBF and WBF if the auto-clock is not used.

The auto-clock is an active HIGH SCL pulse that starts 28 XTAL periods after the SDI read or SDO write via S1BIT. The duration of the auto-clock pulse is 100 XTAL clock periods. If the SCL line is kept LOW by any device that wants to hold up the bus transfer, the auto-clock counter waits after 20 XTAL clock periods so that the auto-clock pulse length will be at least 80 XTAL clock periods (5µs at f<sub>OSC</sub> = 16MHz).

Every bit I/O should be followed by a RBF or WBF bit test. A bit transfer has been finished successfully if after reading a bit the RBF flag is logic 1 or after writing a bit the WBF flag is logic 1. When after reading a bit the RBF flag is still logic 0, the bus status just before the S1SCS status read can be determined as follows:

- If CLH = 0 then a bus device is still stretching the clock.
- If SCL = 1 while CLH = 1 then the SCL pulse is not finished.
- If BB = 0 there has been a STOP condition.

When after writing a bit the WBF flag is still logic 0 and none of the 3 status conditions mentioned for RBF are found then a "bus arbitration lost" condition will be the cause. This can be determined also from the states of the received bit and the last transmitted bit: "arbitration loss" if SDO = 1 and SDI = 0.

**S1SCS: Control and Status Register for the I<sup>2</sup>C-bus**

This register is located at address D8H. Refer to Table 3.

**S1SCS SFR (D8H)****READ**

7	6	5	4	3	2	1	0
SDI	SCI	CLH	BB	RBF	WBF	STR	ENS

**WRITE**

7	6	5	4	3	2	1	0
SDO	SCO	SLH	X	X	X	STR	ENS

**NOTES:**

- SDI and SCI bits:** read-modify-write operations like "SETB bit" or "CLR bit" access SDO and SCO for reading and writing.
- CLH bit:** writing a logic 0 clears this bit, writing a logic 1 has no effect.
- RBF and WBF bits:** writing a logic 0 to CLH also clears these bits.
- X = don't care.

**Table 3. Description of S1SCS Bits**

MNEMONIC	BIT	FUNCTION
SDO/SDI	S1SCS.7	<b>Serial Data Output and the filtered Serial Data Input.</b> SDI data is latched on the rising edge of the filtered serial clock. S1SCS.7 accesses the same memory locations as S1BIT.7. Access of the data bit via S1SCS will not start an auto-clock pulse.
SCO/SCI	S1SCS.6	<b>Serial Clock Output and the filtered Serial Clock Input.</b> Serial clock output SCO is "OR-ed" with the auto-clock. If SCO = 1 the auto-clock output is inhibited. The internal clock stretching logic and external devices can pull the SCL line LOW. If the auto-clock is not used, the SCL line has to be controlled by setting SCO = 1, waiting for CLH = 1 and setting SCO = 0 after the specified SCL HIGH time. (Because of the input filter, CLH will be set at least 8 XTAL clock periods after the SCL LOW-to-HIGH transition.
CLH	S1SCS.5	<b>Serial Clock LOW-to-HIGH transition flag:</b> set with a rising edge of the filtered serial clock. CLH = 1 indicates that, since the last CLH reset, a new valid data bit has been latched in SDI. CLH can be reset by writing a logic 0 to S1SCS.5 or by a read/write of S1BIT. Clearing CLH also clears RBF and WBF.
BB	S1SCS.4	<b>Bus Busy flag:</b> indicating that there has been a START condition that was not yet followed by a STOP condition.
RBF	S1SCS.3	<b>Read Bit Finished flag:</b> indicating a successful bit read. RBF = 1 implies the following conditions: <ul style="list-style-type: none"> <li>– CLH = 1: SCL had a rising edge</li> <li>– SCI = 0: the SCL pulse has finished</li> <li>– SI = 0: no START condition occurred</li> <li>– BB = 1: no STOP condition occurred</li> </ul> The RBF flag can be cleared by clearing the CLH flag.
WBF	S1SCS.2	<b>Write Bit Finished flag:</b> indicating a successful bit write. the same conditions as for RBF are true and also no "arbitration loss" condition occurred. Arbitration is lost if a 1 data bit in SDO was over-ruled on SDA by an external device. The WBF flag can be cleared by clearing the CLH flag.
STR	S1SCS.1	<b>STretch control flag.</b> STR = 1 enabled stretching of all SCL LOW periods. This allows the processor in I <sup>2</sup> C slave mode to react on a fast master. The STR flag remains set until cleared by writing a logic 0 to S1SCS.1.  <b>The STretch (ST) flag</b> (not readable) pulls the serial clock LOW while ST = 1. The ST flag is set on the falling edge of the filtered serial clock if STR = 1. It is also set after reception of a START condition, regardless of the STR contents. ST is cleared with a read or write of S1BIT.
ENS	S1SCS.0	<b>ENable Serial I/O flag.</b> ENS = 1 enables the START detection and clock stretching logic. ENS = 0 can be used to switch off the I <sup>2</sup> C bus hardware. Note that the SDO and SCO control flags must be set to logic 1 before ENS is set to avoid pulling SCL or SDA lines to logic 0.

## 8-bit microcontroller

## P83C524

## INTERRUPT SYSTEM

The P83C524 contains the same interrupt structure as the PCB80C51BH, but with a seven-source interrupt structure with two priority levels.

Table 4. Description of IE Bits

MNEMONIC	BIT	FUNCTION
EA	IE.7	General enable 0 = NO interrupt 1 = ANY interrupt
ES1	IE.6	Enable bit for ET2
ET2	IE.5	Enable Timer 2
ES	IE.4	Enable Serial
ET1	IE.3	Enable Timer 1
EX1	IE.2	Enable External
ET0	IE.1	Enable Timer 0
EX0	IE.0	Enable External

### Table 5. Description of IP Bits

MNEMONIC	BIT	FUNCTION
—	IP.7	Reserved
PS1	IP.6	Bit-level I <sup>2</sup> C
PT2	IP.5	Timer 2 Int
PS	IP.4	Serial Port
PT1	IP.3	Timer 1 int
PX1	IP.2	External In
PT0	IP.1	Timer 0 Int
PX0	IP.0	External In

**IE: Interrupt Enable Register**

This register is located at address A8H. Refer to Table 4.

## IE SFR (A8H)

7	6	5	4	3	2	1	0
EA	ES1	ET2	ES	ET1	EX1	ET0	EX0

**IP:** Interrupt Priority Register

This register is located at address B8H. Refer to Table 5.

IP SFR (B8H)

7	6	5	4	3	2	1	0
-	PS1	PT2	PS	PT1	PX1	PT0	PX0



8-bit microcontroller

P83C524

IDLE AND POWER-DOWN OPERATION

Idle mode operation permits the interrupt, serial ports and timer blocks to function while the CPU is halted. The following functions remain active during Idle mode. These functions may generate an interrupt or reset and thus end the Idle mode:

- Timer 0, Timer 1, Timer 2, Watchdog Timer
- UART, I<sup>2</sup>C-Interface
- External interrupt.

The Power-down operation stops the oscillator. The Power-down mode can only be activated by setting the PD bit in the PCON register.

Idle Mode

The instruction that sets PCON.0 is the last instruction executed in the normal operating mode before Idle mode is activated. Once in the Idle mode, the CPU status is preserved in its entirety: the Stack Pointer, Program Counter, Program Status Word, Accumulator, RAM and all other registers maintain their data during Idle mode. The status of external pins during Idle mode is shown in Table 7.

Power-down Mode

The instruction that sets PCON.1 is the last executed prior to going into the Power-down mode. The oscillator is stopped. Note that the Power-down mode also can be entered when the watchdog has been enabled. The

Power-down mode can be terminated by an external RESET in the same way as in the 80C51 or in addition by any one of the two external interrupts, IE0 or IE1. A reset generated by the WDT terminates the Power-down mode in the same way as an external RESET.

The status of the external pins during Power-down mode is shown in Table 7. If the Power-down mode is activated while in external program memory, the port data that is held in the P2 SFR is restored to Port 2. If the data is a logic 1, the port pin is held HIGH during the Power-down mode by the strong pull-up transistor p1.

Wake-up from Power-down Mode

The Power-down mode of the P83C524 can also be terminated by any one of the two external interrupts, IE0 or IE1. A termination with an external interrupt does not affect the internal data memory and does not affect the Special Function Registers (SFRs).

In order to prevent any interrupt priority problems during wake-up, the priority of the desired wake-up interrupt should be higher than the priorities of all other enabled interrupt sources. The instruction following the one that put the device into the Power-down mode will be the first one which will be executed after an interrupt has been serviced.

Table 6. Internal Registers Status after a RESET

REGISTER	CONTENTS
ACC	00H
B	00H
DPH, DPL	00H
IE	00H
IP	X000 0000B
PCH, PCL	00H
PCON	0XX 0000B
PSW	00H
P0 to P3	FFH
SBUF	Indeterminate
SCON	00H
SP	07H
TCON	00H
TMOD	00H
TH0, TL0	00H
TH1, TL1	00H
T2CON	00H
TH2, TL2	00H
RCAP2H, RCAP2L	00H
S1BIT	X000 0000B
S1INT	0XXX XXXXB
S1SCS	XXX0 0000B
T3	00H
WDCON	A5H

Table 7. Status of the External Pins During Idle and Power-down Modes

MODE	MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle	internal	1	1	port data	port data	port data	port data
Idle	external	1	1	floating	port data	address	port data
Power-down	internal	0	0	port data	port data	port data	port data
Power-down	external	0	0	floating	port data	port data	port data

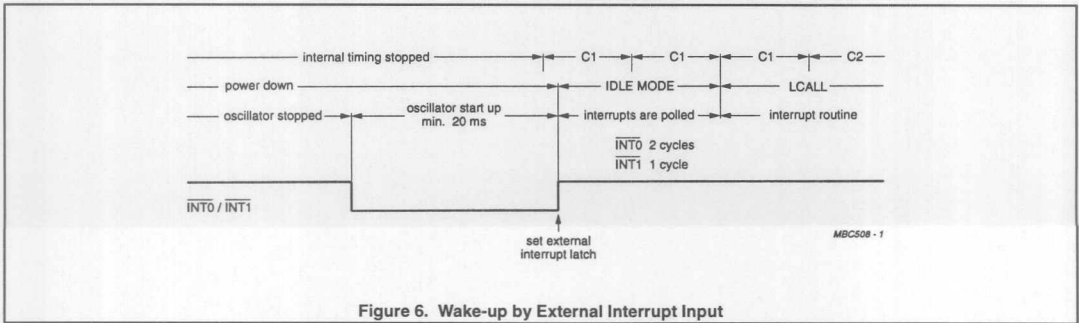


Figure 6. Wake-up by External Interrupt Input

## 8-bit microcontroller

P83C524

## ABSOLUTE MAXIMUM RATINGS

In accordance with the Absolute Maximum Rating System (IEC 134).

SYMBOL	PARAMETER	MIN.	MAX.	UNIT
$V_{DD}$	Supply voltage range	-0.5	+6.5	V
$V_I$	All Input voltages	-0.5	$V_{DD}+0.5$	V
$P_{TOT}$	Total power dissipation	—	1	W
$T_{STG}$	Storage temperature range	-65	+150	°C
$T_{AMB}$	Operating ambient temperature range:			
	version FBx	0	+70	°C
	version FFx	-40	+85	°C
	version FHx	-40	+125	°C

## DC CHARACTERISTICS FBx (0 to +70°C)

 $V_{DD} = 5V \pm 20\%$ ;  $V_{SS} = 0V$ ;  $T_{AMB} = 0$  to +70°C. All voltages with respect to  $V_{SS}$  unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
<b>Supply</b>					
$V_{DD}$	Supply voltage range		4.0	6.0	V
$I_{DD}$	Operating modes	$V_{DD} = 6V$ ; $f_{CLK} = 16MHz$ (notes 1 and 8)	—	35	mA
$I_{ID}$	Idle mode	$V_{DD} = 5V \pm 20\%$ ; $f_{CLK} = 16MHz$ (notes 2 and 8)	—	6	mA
$I_{PD}$	Power-down mode	$2 \leq V_{DD} \leq V_{DDMAX}$ (note 3)	—	100	$\mu A$
<b>Inputs</b>					
$V_{IL}$	LOW level input voltage (except EA, P1.6, P1.7)		-0.5	$0.2V_{DD}-0.1$	V
$V_{IL1}$	LOW level input voltage EA		-0.5	$0.2V_{DD}-0.3$	V
$V_{IL2}$	LOW level input voltage P1.6, P1.7	note 6	-0.5	$0.3V_{DD}$	V
$V_{IH}$	HIGH level input voltage (except RST, XTAL1, P1.6, P1.7)		$0.2V_{DD}+0.9$	$V_{DD}+0.5$	V
$V_{IH1}$	HIGH level input voltage RST, XTAL1		$0.7V_{DD}$	$V_{DD}+0.5$	V
$V_{IH2}$	HIGH level input voltage P1.6, P1.7	note 6	$0.7V_{DD}$	6.0	V
$I_{IL}$	Input current logic 0 Ports 1, 2, and 3 (except P1.6 and P1.7)	$V_I = 0.45V$	—	-50	$\mu A$
$I_{TL}$	Input current logic 1 to logic 0 transition Ports 1, 2, and 3 (except P1.6 and P1.7)	$V_I = 2.0V$	—	-650	$\mu A$
$I_{L1}$	Input leakage current Port 0, EA	$0.45 < V_I < V_{DD}$	—	$\pm 10$	$\mu A$
$I_{L2}$	Input leakage current P1.6 and P1.7	$0V < V_I < 6V$ $0V < V_{DD} < 6V$	—	$\pm 10$	$\mu A$

## 8-bit microcontroller

P83C524

## DC CHARACTERISTICS FBx (0 to +70°C) (Continued)

 $V_{DD} = 5V \pm 20\%$ ;  $V_{SS} = 0V$ ;  $T_{AMB} = 0$  to  $+70^\circ C$ . All voltages with respect to  $V_{SS}$  unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
<b>Outputs</b>					
$V_{OL}$	Low level output voltage Ports 1, 2, and 3 (except P1.6 and P1.7)	$I_{OL} = 1.6mA$ notes 4 and 7	—	0.45	V
$V_{OL1}$	LOW level output voltage Port 0, ALE, PSEN	$I_{OL} = 3.2mA$ notes 4 and 7	—	0.45	V
$V_{OL2}$	LOW level output voltage P1.6 and P1.7	$I_{OL} = 3.0mA$ note 7	—	0.40	V
$V_{OH}$	HIGH level output voltage Ports 1, 2, and 3	$I_{OH} = -60\mu A$ ; $V_{DD} = 5V \pm 10\%$ $I_{OH} = -25\mu A$ $I_{OH} = -10\mu A$	2.4 0.75 $V_{DD}$ 0.9 $V_{DD}$	—	V
$V_{OH1}$	HIGH level output voltage Port 0 in external Bus mode, ALE, PSEN, RST	$I_{OH} = -800\mu A$ ; $V_{DD} = 5V \pm 10\%$ $I_{OH} = -300\mu A$ $I_{OH} = -80\mu A$ note 5	2.4 0.75 $V_{DD}$ 0.9 $V_{DD}$	—	V
$R_{RST}$	RST pull-down resistor		50	150	k $\Omega$
$C_{IO}$	Capacitance of input buffer	Test frequency = 1MHz; $T_{AMB} = 25^\circ C$	—	10	pF

## NOTES TO DC CHARACTERISTICS (FBx):

- The operating supply current is measured with all output pins disconnected; XTAL1 driven with  $t_R = t_F = 5ns$ ;  $V_{IL} = V_{SS} + 0.5V$ ;  $V_{IH} = V_{DD} - 0.5V$ ; XTAL2 not connected;  $E\bar{A} = RST = Port 0 + P1.6 + P1.7 = V_{DD}$ ; the WDT is disabled (by the external RESET).
- The Idle mode supply current is measured with all output pins disconnected; XTAL1 driven with  $t_R = t_F = 5ns$ ;  $V_{IL} = V_{SS} + 0.5V$ ;  $V_{IH} = V_{DD} - 0.5V$ ; XTAL2 not connected; the WDT is disabled;  $E\bar{A} = RST = V_{SS}$ ; Port 0 = P1.6 = P1.7 =  $V_{DD}$ .
- The Power-down current is measured with all output pins disconnected; XTAL2 not connected; WDT is disabled;  $E\bar{A} = RST = XTAL1 = V_{SS}$ ; Port 0 = P1.6 = P1.7 =  $V_{DD}$ .
- Capacitive loading on Port 0 and Port 2 may cause spurious noise pulses to be superimposed on the LOW level output voltage of ALE, Port 1 and Port 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make a 1-to-0 transition during bus operations. In the worst cases (capacitive loading > 100pF), the noise pulse on the ALE line may exceed 0.8V. In such cases it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.
- Capacitive loading on Port 0 and Port 2 may cause the HIGH level output voltage on ALE PSEN to momentarily fall below the 0.9V  $V_{DD}$  specification when the address bits are stabilizing.
- The input threshold voltage of P1.6 and P1.7 (SIO1) meets the I<sup>2</sup>C specification, so a voltage below 0.3  $V_{DD}$  will be recognized as a logic 0 while an input above 0.7  $V_{DD}$  will be recognized as a logic 1.
- Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:
 

Maximum $I_{OL}$ per port pin:	10mA
Maximum $I_{OL}$ per 8-bit port:	
Port 0:	26mA
Ports 1, 2, and 3:	15mA
Maximum total $I_{OL}$ for all output pins:	71mA

 If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
- $I_{DDMAX}$  at other frequencies can be derived from Figure 7, where  $f$  is the external oscillator frequency in MHz;  $I_{DDMAX}$  is given in mA.

$A_{II}$	000-	—	$V_{0.5} = 1V$	input current logic 1 to logic 0 transition (Ports 1, 2, and 3 except P1.6 and P1.7)	$I_{II}$
$A_{II}$	010	—	$0.45 < V < V_{DD}$	input leakage current Port 0, $E\bar{A}$	$I_{II}$
$A_{II}$	010	—	$V_{DD} > V > V_0$ $0.9 < V < V_0$	input leakage current P1.6 and P1.7	$I_{II}$

## 8-bit microcontroller

P83C524

**DC CHARACTERISTICS FFx (–40 to +85°C)**

$V_{DD} = 5V \pm 20\%$ ;  $V_{SS} = 0V$ ;  $T_{AMB} = -40$  to  $+85^\circ\text{C}$  (extended temperature range). All voltages with respect to  $V_{SS}$  unless otherwise specified. DC parameters not included here are the same as for the FBx temperature range data.

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
<b>Inputs</b>					
$V_{IL}$	LOW level input voltage (except EA, P1.6, P1.7)		–0.5	$0.2V_{DD}-0.15$	V
$V_{IL1}$	LOW level input voltage EA		–0.5	$0.2V_{DD}-0.35$	V
$V_{IH}$	HIGH level input voltage (except RST, XTAL1, P1.6, P1.7)		$0.2V_{DD}+1.0$	$V_{DD}+0.5$	V
$V_{IH1}$	HIGH level input voltage RST, XTAL1		$0.7V_{DD}+0.1$	$V_{DD}+0.5$	V
$I_{IL}$	Input current logic 0 Ports 1, 2, and 3 (except P1.6 and P1.7)	$V_I = 0.45V$	–	–75	$\mu\text{A}$
$I_{TL}$	Input current logic 1 to logic 0 transition Ports 1, 2, and 3 (except P1.6 and P1.7)	$V_I = 2.0V$	–	–750	$\mu\text{A}$

**DC CHARACTERISTICS FHx (–40 to +125°C)**

$V_{DD} = 5V \pm 10\%$ ;  $V_{SS} = 0V$ ;  $T_{AMB} = -40$  to  $+125^\circ\text{C}$  (extended temperature range). All voltages with respect to  $V_{SS}$  unless otherwise specified. DC parameters not included here are the same as for the FBx temperature range data.

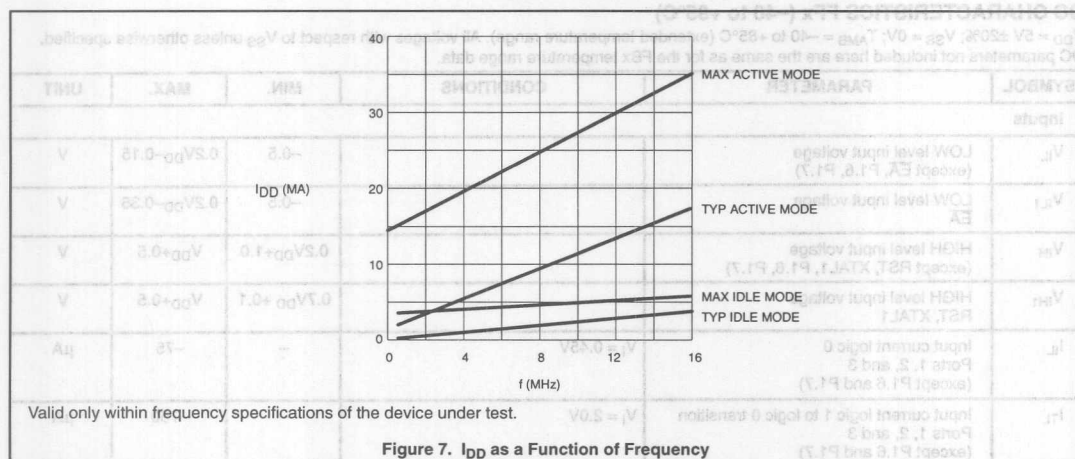
SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
<b>Supply</b>					
$V_{DD}$	Supply voltage range		4.5	5.5	V
$I_{DD}$	Supply current: Operating modes	$V_{DD} = 5.5V$ ; $f_{CLK} = 16\text{MHz}$ (note 1)	–	35	mA
$I_{ID}$	Idle mode	$V_{DD} = 5V \pm 10\%$ ; $f_{CLK} = 16\text{MHz}$ (note 2)	–	6	mA
$I_{PD}$	Power-down mode	$2 \leq V_{PD} \leq V_{DDMAX}$ (note 3)	–	150	$\mu\text{A}$
<b>Inputs</b>					
$V_{IL}$	LOW level input voltage (except EA, P1.6, P1.7)		–0.5	$0.2V_{DD}-0.25$	V
$V_{IL1}$	LOW level input voltage EA		–0.5	$0.2V_{DD}-0.45$	V
$V_{IH}$	HIGH level input voltage (except RST, XTAL1, P1.6, P1.7)		$0.2V_{DD}+1.0$	$V_{DD}+0.5$	V
$V_{IH1}$	HIGH level input voltage RST, XTAL1		$0.7V_{DD}+0.1$	$V_{DD}+0.5$	V
$I_{IL}$	Input current LOW Ports 1, 2, and 3 (except P1.6 and P1.7)	$V_I = 0.45V$	–	–75	$\mu\text{A}$
$I_{TL}$	Input current logic 1 to logic 0 transition Ports 1, 2, and 3 (except P1.6 and P1.7)	$V_I = 2.0V$	–	–750	$\mu\text{A}$

**NOTES:**

1. See notes 1 and 8 of the FBx DC characteristics table.
2. See notes 2 and 8 of the FBx DC characteristics table.
3. See note 3 of the FBx DC characteristics table.

## 8-bit microcontroller

P83C52A



DC CHARACTERISTICS FHX (-40 to +125°C)  
 $V_{DD} = 5V \pm 10\%$ ,  $V_{SS} = 0V$ ,  $T_{AMB} = -40$  to  $+125^\circ C$  (extended temperature range), All voltages with respect to  $V_{SS}$  unless otherwise specified.  
 DC parameters not included here are the same as for the FHX temperature range data.

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
$V_{DD}$	Supply voltage range		4.5	5.5	V
$I_{DD}$	Supply current: Operating modes (note 1)	$V_{DD} = 5V$ , $f_{CLK} = 16MHz$	-	35	mA
$I_{ID}$	Supply current: Idle mode (note 2)	$V_{DD} = 5V \pm 10\%$ , $f_{CLK} = 16MHz$	-	5	mA
$I_{PD}$	Supply current: Power-down mode (note 3)	$5 \leq V_{DD} \leq V_{DDMAX}$	-	150	$\mu A$
$V_{IL}$	LOW level input voltage (except EA, P1.6, P1.7)		-0.5	$0.5V_{DD}-0.35$	V
$V_{LI}$	LOW level input voltage EA		-0.5	$0.5V_{DD}-0.45$	V
$V_{IH}$	HIGH level input voltage (except RST, XTAL1, P1.6, P1.7)		$0.5V_{DD}+1.0$	$V_{DD}+0.5$	V
$V_{HI}$	HIGH level input voltage RST, XTAL1		$0.7V_{DD}+0.1$	$V_{DD}+0.5$	V
$I_L$	input current LOW Ports 1, 2, and 3 (except P1.6 and P1.7)	$V_I = 0.45V$	-	-75	$\mu A$
$I_{L1}$	input current logic 1 to logic 0 transition Ports 1, 2, and 3 (except P1.6 and P1.7)	$V_I = 2.0V$	-	-750	$\mu A$

NOTES:  
 1. See notes 1 and 2 of the FHX DC characteristics table.  
 2. See notes 2 and 3 of the FHX DC characteristics table.  
 3. See note 3 of the FHX DC characteristics table.



## 8-bit microcontroller

P83C524

## AC CHARACTERISTICS

FBX:  $V_{DD} = 5V \pm 20\%$ ;  $V_{SS} = 0V$ ;  $T_{AMB} = 0$  to  $+70^\circ C$ ;  $t_{CK}$  min. = 63ns.FFX:  $V_{DD} = 5V \pm 20\%$ ;  $V_{SS} = 0V$ ;  $T_{AMB} = -40$  to  $+85^\circ C$ ;  $t_{CK}$  min. = 63ns.FHX:  $V_{DD} = 5V \pm 10\%$ ;  $V_{SS} = 0V$ ;  $T_{AMB} = 0$  to  $+125^\circ C$ ;  $t_{CK}$  min. = 63ns.All versions Fxx:  $C_L = 100pF$  for Port 0, ALE and PSEN;  $C_L = 80pF$  for all other outputs unless otherwise specified. $t_{CK}$  min. =  $1/f$  max. (maximum operating frequency);  $t_{CK}$  = clock period.

SYMBOL	PARAMETER	16 MHz		12 MHz		VARIABLE CLOCK		UNIT
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
External program memory								
t <sub>LHLL</sub>	ALE pulse duration	85	—	127	—	2 t <sub>CK</sub> -40	—	ns
t <sub>AVLL</sub>	Address set-up time to ALE	8	—	28	—	t <sub>CK</sub> -55	—	ns
t <sub>LLAX</sub>	Address hold time after ALE	28	—	48	—	t <sub>CK</sub> -35	—	ns
t <sub>LLIV</sub>	Time from ALE to valid instruction input	—	150	—	233	—	4 t <sub>CK</sub> -100	ns
t <sub>LLPL</sub>	Time from ALE to control pulse PSEN	23	—	43	—	t <sub>CK</sub> -40	—	ns
t <sub>PLPH</sub>	Control pulse duration PSEN	143	—	205	—	3 t <sub>CK</sub> -45	—	ns
t <sub>PLIV</sub>	Time from PSEN to valid instruction input	—	83	—	145	—	3 t <sub>CK</sub> -105	ns
t <sub>PXIX</sub>	Input instruction hold time after PSEN	0	—	0	—	0	—	ns
t <sub>PXIZ</sub>	Input instruction float delay after PSEN	—	38	—	59	—	t <sub>CK</sub> -25	ns
t <sub>AVIV</sub>	Address to valid instruction input	—	208	—	312	—	5 t <sub>CK</sub> -105	ns
t <sub>PLAZ</sub>	Address float time to PSEN	—	10	—	10	—	10	ns
External data memory								
t <sub>LHLL</sub>	ALE pulse duration	85	—	127	—	2 t <sub>CK</sub> -40	—	ns
t <sub>AVLL</sub>	Address set-up time to ALE	8	—	28	—	t <sub>CK</sub> -55	—	ns
t <sub>LLAX</sub>	Address hold time after ALE	28	—	48	—	t <sub>CK</sub> -35	—	ns
t <sub>RLRH</sub>	RD pulse duration	275	—	400	—	6 t <sub>CK</sub> -100	—	ns
t <sub>WLWH</sub>	WR pulse duration	275	—	400	—	6 t <sub>CK</sub> -100	—	ns
t <sub>RLDV</sub>	RD to valid data input	—	148	—	252	—	5 t <sub>CK</sub> -165	ns
t <sub>RHDX</sub>	Data hold time after RD	0	—	0	—	0	—	ns
t <sub>RHDZ</sub>	Data float delay after RD	—	55	—	97	—	2 t <sub>CK</sub> -70	ns
t <sub>LLDZ</sub>	Time from ALE to valid data input	—	350	—	517	—	8 t <sub>CK</sub> -150	ns
t <sub>AVDV</sub>	Address to valid data input	—	398	—	585	—	9 t <sub>CK</sub> -165	ns
t <sub>LLWL</sub>	Time from ALE to RD or WR	138	238	200	300	3 t <sub>CK</sub> -50	3 t <sub>CK</sub> +50	ns
t <sub>AVWL</sub>	Time from address to RD or WR	120	—	203	—	4 t <sub>CK</sub> -130	—	ns
t <sub>WHWL</sub>	Time from RD or WR HIGH to ALE HIGH	23	103	43	123	t <sub>CK</sub> -40	t <sub>CK</sub> +40	ns
t <sub>QVWX</sub>	Data valid to WR transition	3	—	23	—	t <sub>CK</sub> -60	—	ns
t <sub>QVWH</sub>	Data set-up time before WR	288	—	433	—	7 t <sub>CK</sub> -150	—	ns
t <sub>WHQX</sub>	Data hold time after WR	13	—	33	—	t <sub>CK</sub> -50	—	ns
t <sub>RLAZ</sub>	Address float delay after RD	—	0	—	0	—	0	ns

## NOTE:

1. The maximum operating frequency is limited to 16MHz and the minimum to 1.2MHz (all versions Fxx).

$t_{LHLL}$	85	—	127	—	$2 t_{CK}-40$	—	ns
$t_{AVLL}$	8	—	28	—	$t_{CK}-55$	—	ns
$t_{LLAX}$	28	—	48	—	$t_{CK}-35$	—	ns
$t_{LLIV}$	—	150	—	233	—	$4 t_{CK}-100$	ns
$t_{LLPL}$	23	—	43	—	$t_{CK}-40$	—	ns
$t_{PLPH}$	143	—	205	—	$3 t_{CK}-45$	—	ns
$t_{PLIV}$	—	83	—	145	—	$3 t_{CK}-105$	ns
$t_{PXIX}$	0	—	0	—	0	—	ns
$t_{PXIZ}$	—	38	—	59	—	$t_{CK}-25$	ns
$t_{AVIV}$	—	208	—	312	—	$5 t_{CK}-105$	ns
$t_{PLAZ}$	—	10	—	10	—	10	ns

## 8-bit microcontroller

P83C524

I<sup>2</sup>C CHARACTERISTICS (bit-level)

SYMBOL	PARAMETER	INPUT	OUTPUT	I <sup>2</sup> C SPEC.	UNIT
<b>SCL Timing</b>					
t <sub>HD;STA</sub>	START condition hold time	≥ 14 t <sub>CK</sub> ; note 1	note 2	≥ 4.0	μs
t <sub>LOW</sub>	SCL LOW time	≥ 16 t <sub>CK</sub>	note 2	≥ 4.7	μs
t <sub>HIGH</sub>	SCL HIGH time	≥ 14 t <sub>CK</sub> ; note 1	≥ 80 t <sub>CK</sub> ; note 3	≥ 4.0	μs
t <sub>RC</sub>	SCL RISE time	≤ 1; note 4	note 5	≤ 1.0	μs
t <sub>FC</sub>	SCL FALL time	≤ 0.3; note 4	≤ 0.3; note 6	≤ 0.3	μs
<b>SDA Timing</b>					
t <sub>SU;DAT</sub>	Data set-up time	≥ 250 ns	note 2	≥ 250	ns
t <sub>HD;DAT</sub>	Data hold time	≥ 0 ns	note 2	≥ 0	ns
t <sub>SU;STA</sub>	Repeated START set-up time	≥ 14 t <sub>CK</sub> ; note 1	note 2	≥ 4.7	μs
t <sub>SU;STO</sub>	STOP condition set-up time	≥ 14 t <sub>CK</sub> ; note 1	note 2	≥ 4.0	μs
t <sub>BUF</sub>	Bus free time	≥ 14 t <sub>CK</sub> ; note 1	note 2	≥ 4.7	μs
t <sub>RD</sub>	SDA RISE time	≤ 1; note 4	note 5	≤ 1.0	μs
t <sub>FD</sub>	SDA FALL time	≤ 300ns; note 4	≤ 0.3; note 6	≤ 0.3	μs

## NOTES:

1. At f<sub>CLK</sub> = 3.5MHz, this evaluates to 14 × 286ns = 4μs, i.e., the bit-level I<sup>2</sup>C interface can respond to the I<sup>2</sup>C protocol for f<sub>CLK</sub> ≥ 3.5MHz.
2. This parameter is determined by the user software, it has to comply with the I<sup>2</sup>C specification.
3. This value gives the auto-clock pulse length which meets the I<sup>2</sup>C specification for the specified XTAL1 clock frequency range. Alternatively, the SCL pulse may be timed by software.
4. Spikes on SDA and SCL lines with a duration of less than 4 × f<sub>CLK</sub> will be filtered out.
5. The RISE time is determined by the external bus line capacitance and pull-up resistor, it must be ≤ 1 μs.
6. The maximum capacitance on bus lines SDA and SCL is 400 pF.

## XTAL1 CHARACTERISTICS

Oscillator circuitry: crystal capacitors: C1 = C2 = 20pF (see Figure 13).

Table 8. External Clock Drive XTAL

SYMBOL	PARAMETER	VARIABLE CLOCK F = 1.2 TO 16 MHz		UNIT
		MIN.	MAX.	
f <sub>CLK</sub>	Clock frequency	1.2	16	MHz
t <sub>CK</sub>	Clock period	63	833	ns
t <sub>HIGH</sub>	HIGH time	20	t <sub>CK</sub> - t <sub>LOW</sub>	ns
t <sub>LOW</sub>	LOW time	20	t <sub>CK</sub> - t <sub>HIGH</sub>	ns
t <sub>R</sub>	RISE time	—	20	ns
t <sub>F</sub>	FALL time	—	20	ns
t <sub>CY</sub>	Cycle time (t <sub>CY</sub> = 12 t <sub>CK</sub> )	0.75	10	ns

## SERIAL PORT CHARACTERISTICS

See Table 9 and Figure 14.

Table 9. External Clock Drive XTAL

V<sub>DD</sub> = 5V ±20%; V<sub>SS</sub> = 0V; T<sub>AMB</sub> = 0 to +70°C; Load Capacitance = 80pF.

SYMBOL	PARAMETER	12MHz OSCILLATOR		VARIABLE OSCILLATOR		UNIT
		MIN.	MAX.	MIN.	MAX.	
t <sub>XLXL</sub>	Serial Port clock cycle time	1	—	12 t <sub>CK</sub>	—	μs
t <sub>QVXH</sub>	Output data setup to clock rising edge	700	—	10 t <sub>CK</sub> -133	—	ns
t <sub>XHQX</sub>	Output data hold after clock rising edge	50	—	2 t <sub>CK</sub> -117	—	ns
t <sub>XHDX</sub>	Input data hold after clock rising edge	0	—	0	—	ns
t <sub>XHDV</sub>	Clock rising edge to input data valid	—	700	—	10 t <sub>CK</sub> -133	ns

## 8-bit microcontroller

P83C524

## TIMING DIAGRAMS

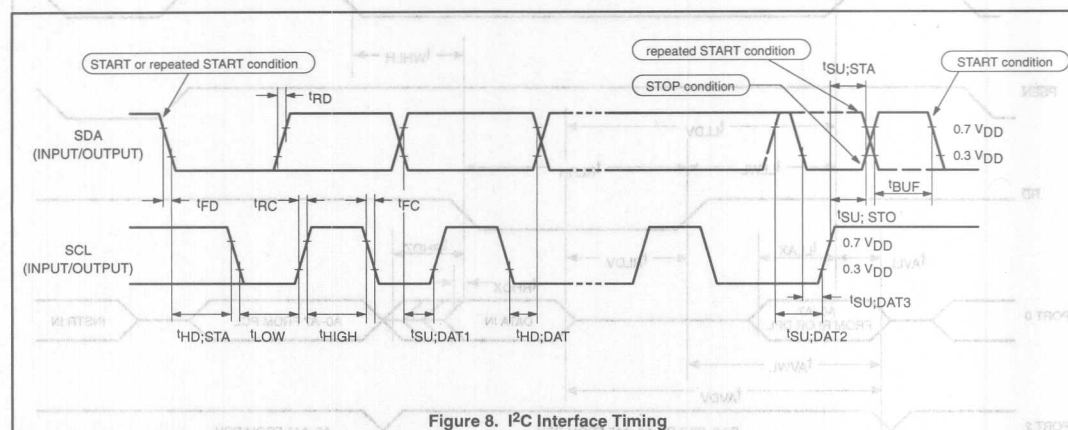


Figure 8. I²C Interface Timing

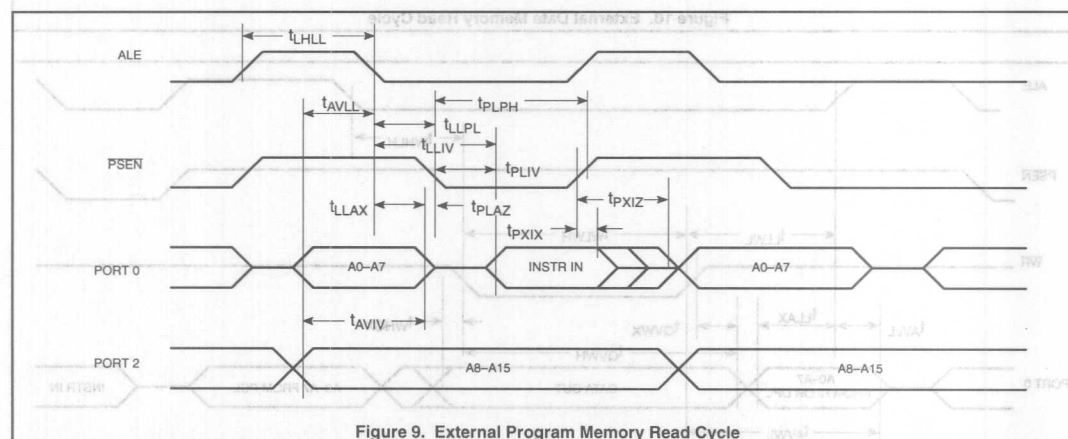


Figure 9. External Program Memory Read Cycle

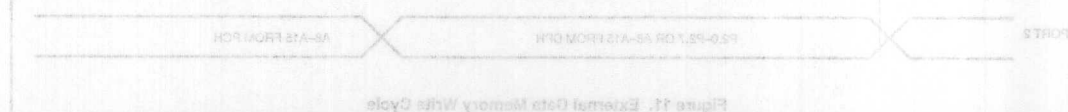


Figure 10. External Data Memory Read Cycle

## 8-bit microcontroller

P83C524

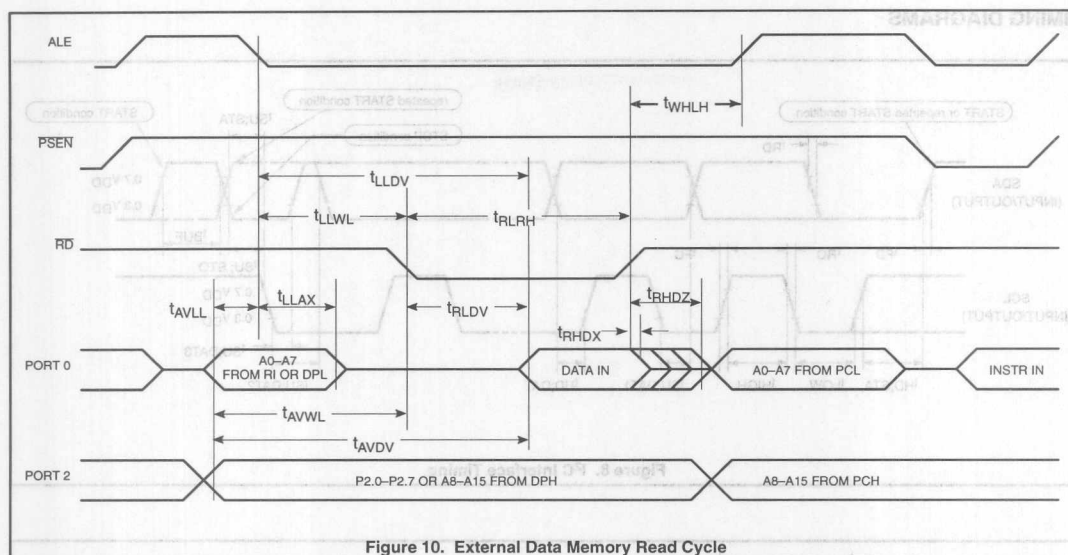


Figure 10. External Data Memory Read Cycle

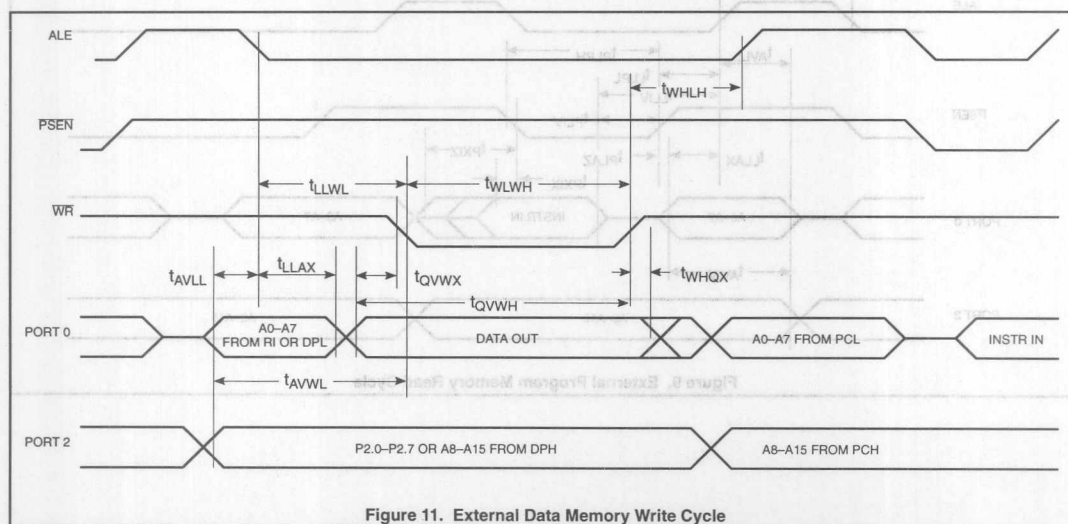
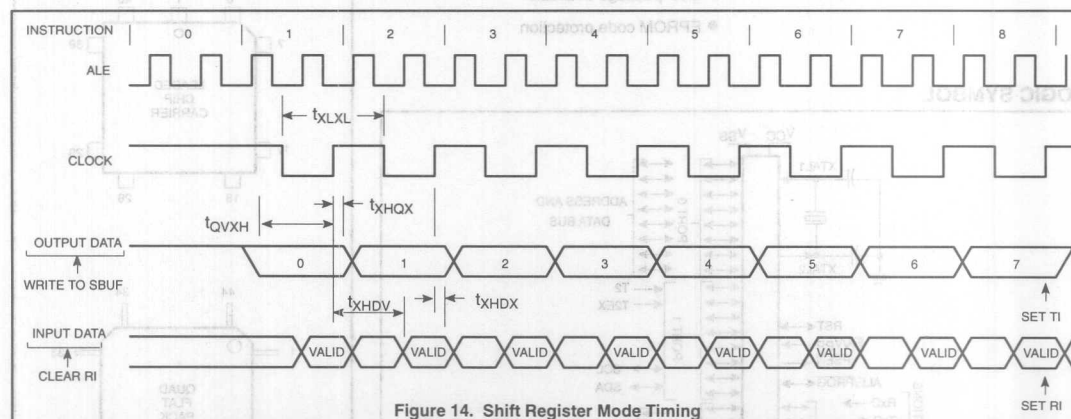
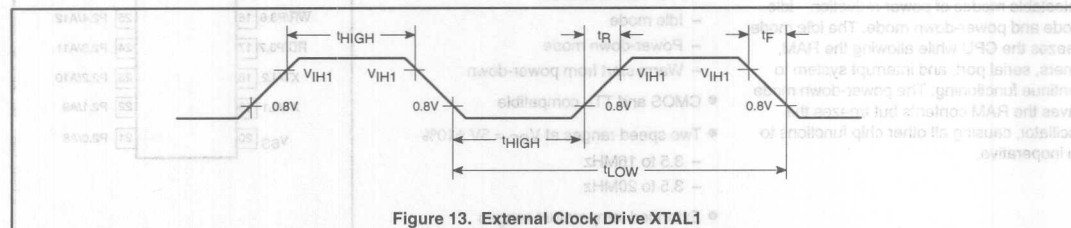
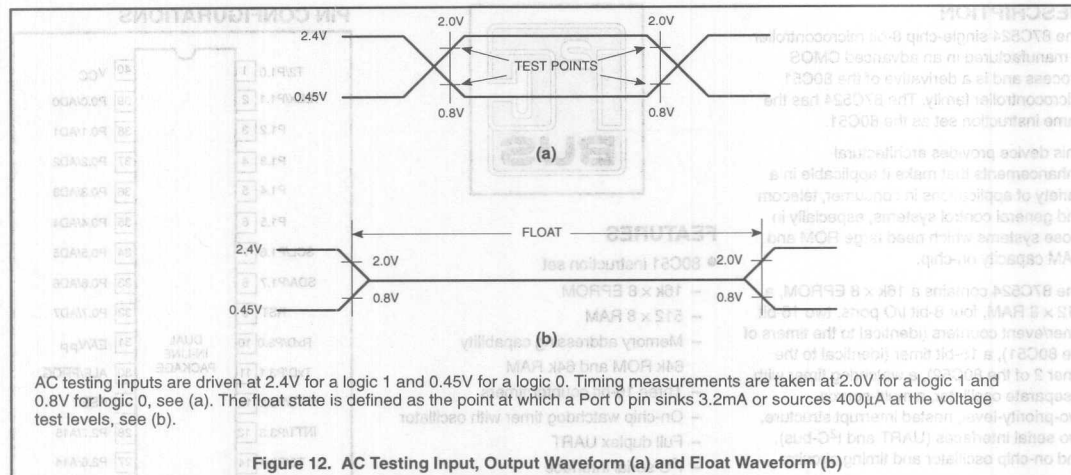


Figure 11. External Data Memory Write Cycle

## 8-bit microcontroller

P83C524





## CMOS single-chip 8-bit microcontroller

87C524

## DESCRIPTION

The 87C524 single-chip 8-bit microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 87C524 has the same instruction set as the 80C51.

This device provides architectural enhancements that make it applicable in a variety of applications in consumer, telecom and general control systems, especially in those systems which need large ROM and RAM capacity on-chip.

The 87C524 contains a  $16k \times 8$  EPROM, a  $512 \times 8$  RAM, four 8-bit I/O ports, two 16-bit timer/event counters (identical to the timers of the 80C51), a 16-bit timer (identical to the timer 2 of the 80C52), a watchdog timer with a separate oscillator, a multi-source, two-priority-level, nested interrupt structure, two serial interfaces (UART and I<sup>2</sup>C-bus), and on-chip oscillator and timing circuits.

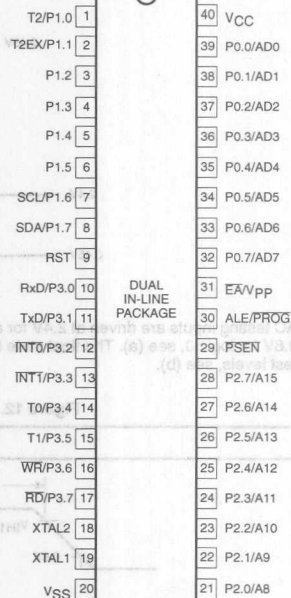
In addition, the 87C524 has two software selectable modes of power reduction—idle mode and power-down mode. The idle mode freezes the CPU while allowing the RAM, timers, serial port, and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.



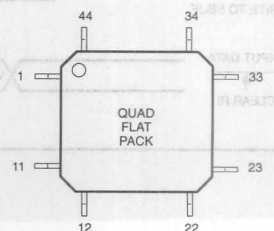
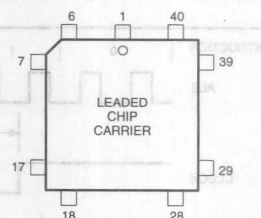
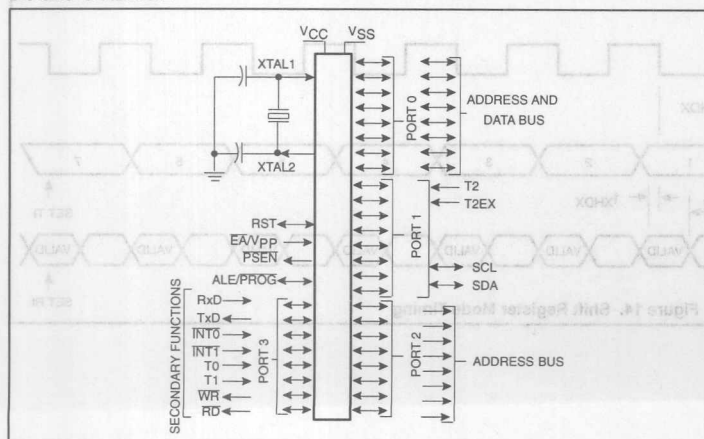
## FEATURES

- 80C51 instruction set
  - $16k \times 8$  EPROM
  - $512 \times 8$  RAM
  - Memory addressing capability  $64k$  ROM and  $64k$  RAM
  - Three 16-bit counter/timers
  - On-chip watchdog timer with oscillator
  - Full duplex UART
  - I<sup>2</sup>C serial interface
- Power control modes:
  - Idle mode
  - Power-down mode
  - Warm start from power-down
- CMOS and TTL compatible
- Two speed ranges at  $V_{CC} = 5V \pm 10\%$ 
  - 3.5 to 16MHz
  - 3.5 to 20MHz
- Extended temperature ranges
- OTP package available
- EPROM code protection

## PIN CONFIGURATIONS



## LOGIC SYMBOL

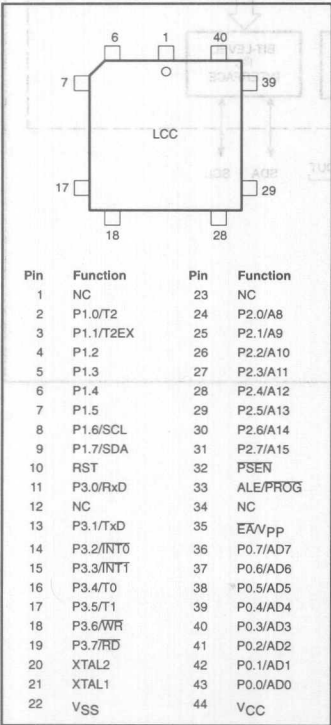


CMOS single-chip 8-bit microcontroller 87C524

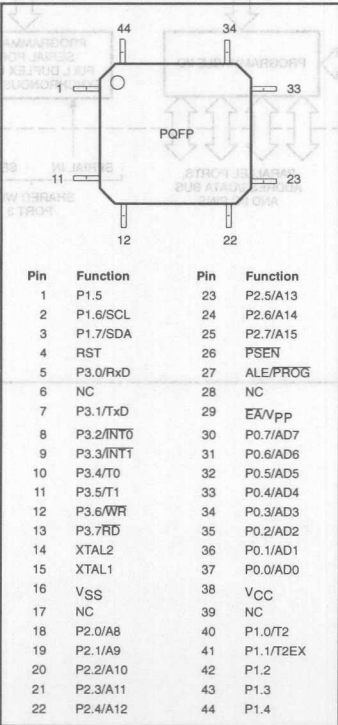
ORDERING INFORMATION

EPROM	TEMPERATURE RANGE °C AND PACKAGE	FREQUENCY	DRAWING NUMBER
P87C524EBP N	0 to +70, Plastic Dual In-line Package	16MHz	0415C
P87C524EBF FA	0 to +70, Ceramic Dual In-line Package w/Window	16MHz	0590B
P87C524EBA A	0 to +70, Plastic Leaded Chip Carrier	16MHz	0403G
P87C524EBL KA	0 to +70, Ceramic Leaded Chip Carrier w/Window	16MHz	1472A
P87C524EBB B	0 to +70, Plastic Quad Flat Pack	16MHz	SOT311
P87C524GFP N	-40 to +85, Plastic Dual In-line Package	20MHz	0415C
P87C524GFF FA	-40 to +85, Ceramic Dual In-line Package w/Window	20MHz	0590B
P87C524GFA A	-40 to +85, Plastic Leaded Chip Carrier	20MHz	0403G
P87C524GFL KA	-40 to +85, Ceramic Leaded Chip Carrier w/Window	20MHz	1472A

CERAMIC AND PLASTIC LEADED  
CHIP CARRIER PIN FUNCTIONS

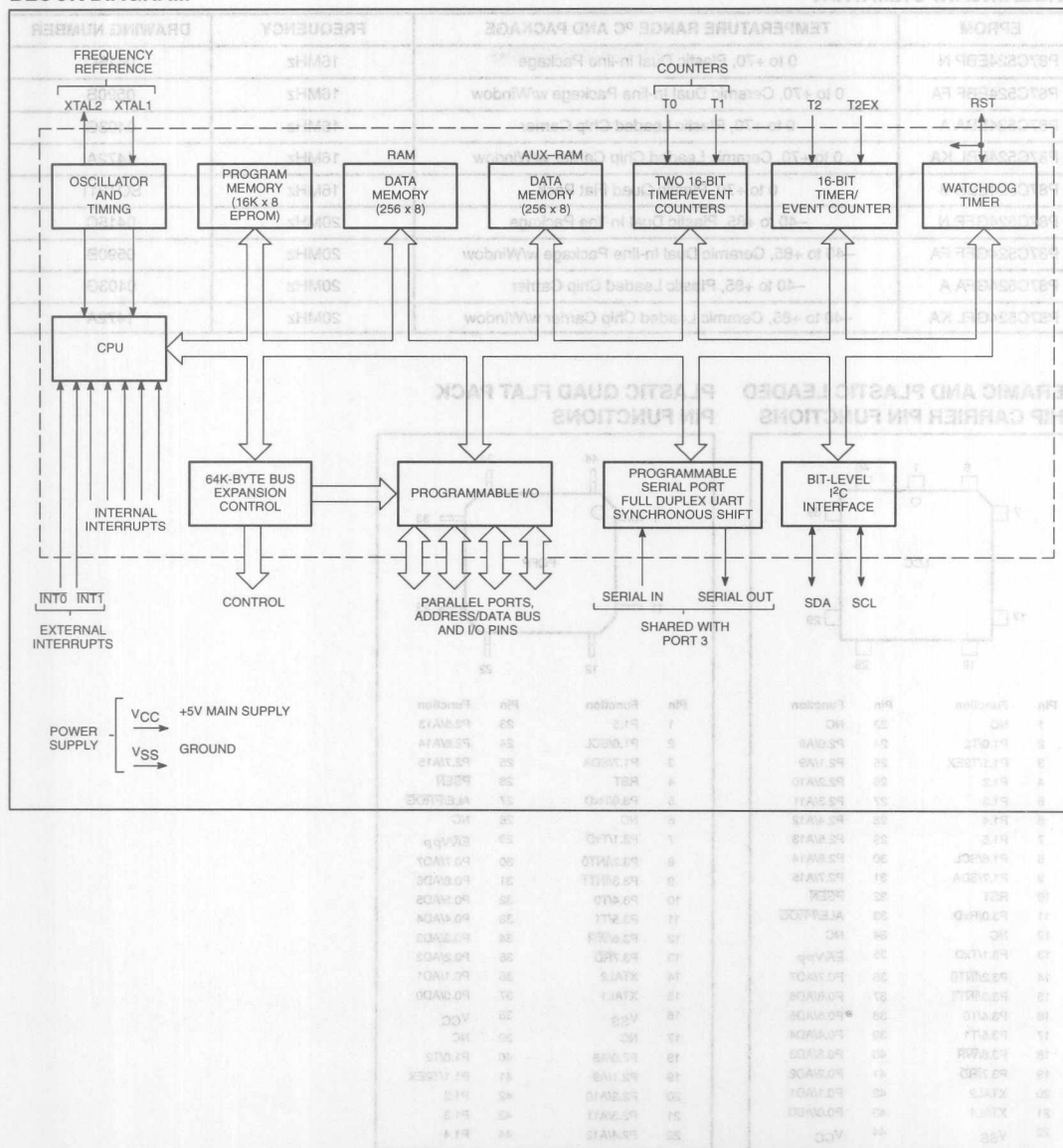


PLASTIC QUAD FLAT PACK  
PIN FUNCTIONS



## CMOS single-chip 8-bit microcontroller 87C52A

## BLOCK DIAGRAM



## CMOS single-chip 8-bit microcontroller 87C524

## PIN DESCRIPTION

MNEMONIC	PIN NO.			TYPE	NAME AND FUNCTION
	DIP	LCC	QFP		
V <sub>SS</sub>	20	22	16	I	<b>Ground:</b> 0V reference.
V <sub>CC</sub>	40	44	38	I	<b>Power Supply:</b> This is the power supply voltage for normal, idle, and power-down operation.
P0.0–0.7	39–32	43–36	37–30	I/O	<b>Port 0:</b> Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s. Port 0 also outputs the code bytes during program verification in the P87C524. External pull-ups are required during program verification.
P1.0–P1.7	1–8	2–9	40–44, 1–3	I/O	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pull-ups, except P1.6 and P1.7 which are open drain. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 1 can sink/source one TTL (4LSTTL) inputs. Port 1 receives the low-order address byte during program memory verification. Port 1 also serves alternate functions for timer 2: <b>T2 (P1.0):</b> Timer/counter 2 external count input. <b>T2EX (P1.1):</b> Timer/counter 2 trigger input. <b>SCL (P1.6):</b> I <sup>2</sup> C serial port clock line. <b>SDA (P1.7):</b> I <sup>2</sup> C serial port data line.
P2.0–P2.7	21–28	24–31	18–25	I/O	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register.
P3.0–P3.7	10–17	11, 13–19	5, 7–13	I/O	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 3 also serves the special features of the SC80C51 family, as listed below: <b>RxD (P3.0):</b> Serial input port <b>TxD (P3.1):</b> Serial output port <b>INT0 (P3.2):</b> External interrupt <b>INT1 (P3.3):</b> External interrupt <b>T0 (P3.4):</b> Timer 0 external input <b>T1 (P3.5):</b> Timer 1 external input <b>WR (P3.6):</b> External data memory write strobe <b>RD (P3.7):</b> External data memory read strobe
RST	9	10	4	I/O	<b>Reset:</b> A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to V <sub>SS</sub> permits a power-on reset using only an external capacitor to V <sub>CC</sub> . After a watchdog timer overflow, this pin is pulled high while the internal reset signal is active.
ALE/PROG	30	33	27	I/O	<b>Address Latch Enable/Program Pulse:</b> Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory. This pin is also the program pulse input (PROG) during EPROM programming.
PSEN	29	32	26	O	<b>Program Store Enable:</b> The read strobe to external program memory. When the device is executing code from the external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory. PSEN is not activated during fetches from internal program memory.
EA/V <sub>PP</sub>	31	35	29	I	<b>External Access Enable/Programming Supply Voltage:</b> EA must be externally held low to enable the device to fetch code from external program memory locations 0000H to 7FFFH. If EA is held high, the device executes from internal program memory unless the program counter contains an address greater than 7FFFH. This pin also receives the 12.75V programming supply voltage (V <sub>PP</sub> ) during EPROM programming.
XTAL1	19	21	15	I	<b>Crystal 1:</b> Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	20	14	O	<b>Crystal 2:</b> Output from the inverting oscillator amplifier.

## CMOS single-chip 8-bit microcontroller 87C524

Table 1. 8XC524/8XC528 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB						LSB		
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
DPTR:	Data pointer (2 bytes):										
DPH	Data pointer high	83H									00H
DPL	Data pointer low	82H									00H
IE*#	Interrupt enable	A8H	AF	AE	AD	AC	AB	AA	A9	A8	
			EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0	00H
			BF	BE	BD	BC	BB	BA	B9	B8	
IP*#	Interrupt priority	B8H	—	PS1	PT2	PS0	PT1	PX1	PT0	PX0	x0000000B
			87	86	85	84	83	82	81	80	
P0*	Port 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
			97	96	95	94	93	92	91	90	
P1*	Port 1	90H	SDA	SEL	—	—	—	—	T2EX	T2	FFH
			A7	A6	A5	A4	A3	A2	A1	A0	
P2*	Port 2	A0H	A15	A14	A13	A12	A11	A10	A9	A8	FFH
			B7	B6	B5	B4	B3	B2	B1	B0	
P3*	Port 3	B0H	RD	WR	T1	T0	INT1	INT0	TxD	RxD	FFH
PCON	Power control	87H	SMOD	—	—	—	GF1	GF0	PD	IDL	0xxx0000B
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program status word	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	00H
RCAP2H#	Capture high	CBH									00H
RCAP2L#	Capture low	CAH									00H
SBUF	Serial data buffer	99H									xxxxxxxB
			9F	9E	9D	9C	9B	9A	99	98	
SCON*	Serial controller	98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00H
S1BIT#	Serial I <sup>2</sup> C data	D9H/RD	SDI	0	0	0	0	0	0	0	x0000000B
		WR	SD0	X	X	X	X	X	X	X	0xxxxxxxB
S1INT#	Serial I <sup>2</sup> C interrupt	DAH	INT	X	X	X	X	X	X	X	0xxxxxxxB
			DF	DE	DD	DC	DB	DA	D9	D8	
S1SCS*#	Serial I <sup>2</sup> C control	D8H/RD	SDI	SCI	CLH	BB	RBF	WBF	STR	ENS	xxxx0000B
		WR	SD0	SC0	CLH	X	X	X	STR	ENS	00xxxx00B
SP	Stack pointer	81H									07H
			8F	8E	8D	8C	8B	8A	89	88	
TCON*	Timer control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
			CF	CE	CD	CC	CB	CA	C9	C8	
T2CON*#	Timer 2 control	C8H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	00H
TH0	Timer high 0	8CH									00H
TH1	Timer high 1	8DH									00H
TH2#	Timer high 2	CDH									00H
TL0	Timer low 0	8AH									00H
TL1	Timer low 1	8BH									00H
TL2#	Timer low 2	CCH									00H
T3#	Watchdog timer	FFH									00H
TMOD	Timer mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H
WDCON#	Watchdog control	A5H									A5H

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.



## CMOS single-chip 8-bit microcontroller 87C524

Table 2. Internal and External Program Memory Access with Security Bit Set

INSTRUCTION	ACCESS TO INTERNAL PROGRAM MEMORY	ACCESS TO EXTERNAL PROGRAM MEMORY
MOVC in internal program memory	YES	YES
MOVC in external program memory	NO	YES

**INTERNAL DATA MEMORY**

The internal data memory is divided into three physically separated segments: 256 bytes of RAM, 256 bytes of AUX-RAM, and a 128 bytes special function area. These can be addressed each in a different way.

- RAM 0 to 127 can be addressed directly and indirectly as in the 80C51. Address pointers are R0 and R1 of the selected register bank.
- RAM 128 to 255 can only be addressed indirectly as in the 80C51. Address pointers are R0 and R1 of the selected register bank.
- AUX-RAM 0 to 255 is indirectly addressed in the same way as external data memory with the MOVX instructions. Address pointers are R0, R1 of the selected register bank and DPTR. An access to AUX-RAM 0 to 255 will not affect ports P0, P2, P3.6 and P3.7.

An access to external data memory locations higher than 255 will be performed with the MOVX DPTR instructions in the same way as in the 8051 structure, so with P0 and P2 as data/address bus and P3.6 and P3.7 as write and read timing signals. Note that these external data memory cannot be accessed with R0 and R1 as address pointer.

**TIMER 2**

Timer 2 is functionally equal to the Timer 2 of the 8052AH. Timer 2 is a 16-bit timer/counter. These 16 bits are formed by two special function registers TL2 and TH2. Another pair of special function register RCAP2L and RCAP2H form a 16-bit capture register or a 16-bit reload register. Like Timer 0 and 1, it can operate either as a timer or as an event counter. This is selected by bit C/T2N in the special function register T2CON. It has three operating modes: capture, autoloader, and baud rate generator mode which are selected by bits in T2CON.

**WATCHDOG TIMER T3**

The watchdog timer consists of an 11-bit prescaler and an 8-bit timer formed by special function register T3. The prescaler is incremented by an on-chip oscillator with a fixed frequency of 1MHz. The maximum tolerance on this frequency is -50% and +100%. The 8-bit timer increments every 2048 cycles of the on-chip oscillator. When a timer overflow occurs, the microcontroller is

reset and a reset output pulse of  $16 \times 2048$  cycles of the on-chip oscillator is generated at pin RST. The internal RESET signal is not inhibited when the external RST pin is kept low by, for example, an external reset circuit. The RESET signal drives port 1, 2, 3 into the high state and port 0 into the high impedance state.

The watchdog timer is controlled by one special function register WDCON with the direct address location A5H. WDCON can be read and written by software. A value of A5H in WDCON halts the on-chip oscillator and clears both the prescaler and timer T3. After the RESET signal, WDCON contains A5H. Every value other than A5H in WDCON enables the watchdog timer. When the watchdog timer is enabled, it runs independently of the XTAL-clock.

Timer T3 can be read on the fly. Timer T3 can only be written if WDCON contains the value 5AH. A successful write operation to T3 will clear the prescaler and WDCON, leaving the watchdog enabled and preventing inadvertent changes of T3. To prevent an overflow of the watchdog timer, the user program has to reload the watchdog timer within periods that are shorter than the programmed watchdog timer interval. This time interval is determined by an 8-bit value that has to be loaded in register T3 while at the same time the prescaler is cleared by hardware.

$$\text{Watchdog timer interval} = \frac{[256 - (T3)] \times 2048}{\text{on-chip oscillator frequency}}$$

**BIT-LEVEL I<sup>2</sup>C INTERFACE**

This bit-level serial I/O interface supports the I<sup>2</sup>C-bus. P1.6/SCL and P1.7/SDA are the serial I/O pins. These two pins meet the I<sup>2</sup>C specification concerning the input levels and output drive capability. Consequently, these pins have an open drain output configuration. All the four modes of the I<sup>2</sup>C-bus are supported:

- master transmitter
- master receiver
- slave transmitter
- slave receiver

The advantages of the bit-level I<sup>2</sup>C hardware compared with a full software I<sup>2</sup>C implementation are:

- the hardware can generate the SCL pulse

- Testing a single bit (RBF respectively, WBF) is sufficient as a check for error free transmission.

The bit-level I<sup>2</sup>C hardware operates on serial bit level and performs the following functions:

- filtering the incoming serial data and clock signals
- recognizing the START condition
- generating a serial interrupt request SI after reception of a START condition and the first falling edge of the serial clock
- recognizing the STOP condition
- recognizing a serial clock pulse on the SCL line
- latching a serial bit on the SDA line (SDI)
- stretching the SCL LOW period of the serial clock to suspend the transfer of the next serial data bit
- setting Read Bit Finished (RBF) when the SCL clock pulse has finished and Write Bit Finished (WBF) if there is no arbitration loss detected (i.e., SDA = 0 while SDO = 1)
- setting a serial clock Low-to-High detected (CLH) flag
- setting a Bus Busy (BB) flag on a START condition and clearing this flag on a STOP condition
- releasing the SCL line and clearing the CLH, RBF and WBF flags to resume transfer of the next serial data bit
- generating an automatic clock if the single bit data register S1BIT is used in master mode.

The following functions must be done in software:

- handling the I<sup>2</sup>C START interrupts
- converting serial to parallel data when receiving
- converting parallel to serial data when transmitting
- comparing the received slave address with its own
- interpreting the acknowledge information
- guarding the I<sup>2</sup>C status if RBF or WBF = 0.

Additionally, if acting as master:

- generating START and STOP conditions
- handling bus arbitration
- generating serial clock pulses if S1BIT is not used.

Three SFRs control the bit-level I<sup>2</sup>C interface: S1INT, S1BIT and S1SCS.

## CMOS single-chip 8-bit microcontroller 87C52A

**OSCILLATOR CHARACTERISTICS**

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator, as shown in the logic symbol, page 460.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

**RESET**

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. To insure a good power-up reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few milliseconds) plus two machine cycles. At power-up, the voltage on  $V_{CC}$  and RST must come up at the same time for a proper start-up.

**IDLE MODE**

In idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

**POWER-DOWN MODE**

In the power-down mode, the oscillator is stopped and the instruction to invoke power-down is the last instruction executed. The power-down mode can be terminated by a RESET in the same way as in the 80C51 or in addition by one of two external interrupts, INT0 or INT1. A termination with an external interrupt does not affect the internal data memory and does not affect the special function registers. This makes it possible to exit power-down without changing the port output levels. To terminate the power-down mode with an external interrupt INT0 or INT1 must be switched to level-sensitive and must be enabled. The external interrupt input

signal INT0 and INT1 must be kept low until the oscillator has restarted and stabilized. An instruction following the instruction that puts the device in the power-down mode will be executed. A reset generated by the watchdog timer terminates the power-down mode in the same way as an external RESET, and only the contents of the on-chip RAM are preserved. The control bits for the reduced power modes are in the special function register PCON.

**DESIGN CONSIDERATIONS**

At power-on, the voltage on  $V_{CC}$  and RST must come up at the same time for a proper start-up.

When the idle mode is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write when idle is terminated by reset, the instruction following the one that invokes idle should not be one that writes to a port pin or to external memory.

Table 3 shows the state of I/O ports during low current operating modes.

**Table 3. External Pin Status During Idle and Power-Down Modes**

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

**ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>**

PARAMETER	RATING	UNIT
Operating temperature under bias	0 to +70 or -40 to +85	°C
Storage temperature range	-65 to +150	°C
Voltage on EA/ $V_{PP}$ pin to $V_{SS}$	0 to +13.0	V
Voltage on any other pin to $V_{SS}$	-0.5 to $V_{CC} + 0.5$	V
Input, output current on any two pins	±10	mA
Power dissipation (based on package heat transfer limitations, not device power consumption)	1.0	W

**NOTES:**

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
- This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
- Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to  $V_{SS}$  unless otherwise noted.

## CMOS single-chip 8-bit microcontroller 87C524

## DC ELECTRICAL CHARACTERISTICS

$T_{amb} = 0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$  or  $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$

SYMBOL	PARAMETER	PART TYPE	TEST CONDITIONS	LIMITS		UNIT
				MIN	MAX	
$V_{IL}$	Input low voltage, except EA, P1.6/SCL, P1.7/SDA	0 to $+70^{\circ}\text{C}$ $-40$ to $+85^{\circ}\text{C}$		-0.5 -0.5	$0.2V_{CC}-0.1$ $0.2V_{CC}-0.15$	V V
$V_{IL1}$	Input low voltage to EA	0 to $+70^{\circ}\text{C}$ $-40$ to $+85^{\circ}\text{C}$		0 0	$0.2V_{CC}-0.3$ $0.2V_{CC}-0.35$	V V
$V_{IL2}$	Input low voltage to P1.6/SCL, P1.7/SDA <sup>5</sup>			-0.5	1.5	V
$V_{IH}$	Input high voltage, except XTAL1, RST, P1.6/SCL, P1.7/SDA	0 to $+70^{\circ}\text{C}$ $-40$ to $+85^{\circ}\text{C}$		$0.2V_{CC}+0.9$ $0.2V_{CC}+1$	$V_{CC}+0.5$ $V_{CC}+0.5$	V V
$V_{IH1}$	Input high voltage, XTAL1, RST	0 to $+70^{\circ}\text{C}$ $-40$ to $+85^{\circ}\text{C}$		$0.7V_{CC}$ $0.7V_{CC}+0.1$	$V_{CC}+0.5$ $V_{CC}+0.5$	V V
$V_{IH2}$	Input high voltage, P1.6/SCL, P1.7/SDA <sup>5</sup>			3.0	6.0	V
$V_{OL}$	Output low voltage, ports 1, 2, 3, except P1.6/SCL, P1.7/SDA		$I_{OL} = 1.6\text{mA}^1$		0.45	V
$V_{OL1}$	Output low voltage, port 0, ALE, PSEN		$I_{OL} = 3.2\text{mA}^1$		0.45	V
$V_{OL2}$	Output low voltage, P1.6/SCL, P1.7/SDA		$I_{OL} = 3.0\text{mA}^1$		0.4	V
$V_{OH}$	Output high voltage, ports 1, 2, 3 <sup>2</sup>		$I_{OH} = -60\mu\text{A}$ $I_{OH} = -25\mu\text{A}$ $I_{OH} = -10\mu\text{A}$	2.4 $0.75V_{CC}$ $0.9V_{CC}$		V V V
$V_{OH1}$	Output high voltage, Port 0 in external bus mode, ALE, PSEN, RST		$I_{OH} = -800\mu\text{A}$ $I_{OH} = -300\mu\text{A}$ $I_{OH} = -80\mu\text{A}$	2.4 $0.75V_{CC}$ $0.9V_{CC}$		V V V
$I_{IL}$	Logical 0 input current, ports 1, 2, 3, except P1.6/SCL, P1.7/SDA	0 to $+70^{\circ}\text{C}$ $-40$ to $+85^{\circ}\text{C}$	$V_{IN} = 0.45\text{V}$		-50 -75	$\mu\text{A}$ $\mu\text{A}$
$I_{TL}$	Logical 1-to-0 transition current, ports 1, 2, 3, except P1.6/SCL, P1.7/SDA	0 to $+70^{\circ}\text{C}$ $-40$ to $+85^{\circ}\text{C}$	See Note 3		-650 -750	$\mu\text{A}$ $\mu\text{A}$
$I_{L1}$	Input leakage current, port 0		$V_{IN} = V_{IL}$ or $V_{IH}$		$\pm 10$	$\mu\text{A}$
$I_{L2}$	Input leakage current, P1.6/SCL, P1.7/SDA		$0\text{V} < V_i < 6\text{V}$ $0\text{V} < V_{CC} < 5.5\text{V}$		$\pm 10$	$\mu\text{A}$
$I_{CC}$	Power supply current: Active mode @ 16MHz Idle mode @ 16MHz Power down mode	0 to $+70^{\circ}\text{C}$ $-40$ to $+85^{\circ}\text{C}$ 0 to $+70^{\circ}\text{C}$ $-40$ to $+85^{\circ}\text{C}$	See Note 4		25 35 5 6 50	mA mA mA mA $\mu\text{A}$
$R_{RST}$	Internal reset pull-down resistor			50	300	k $\Omega$
$C_{IO}$	Pin capacitance				10	pF

## NOTES:

- Capacitive loading on ports 0 and 2 may cause spurious noise to be superimposed on the  $V_{OL}$ s of ALE and ports 1 and 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading  $> 100\text{pF}$ ), the noise pulse on the ALE pin may exceed 0.8V. In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input. Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows: 10mA per port pin, port 0 total (all bits) 26mA, ports 1, 2, and total each (all bits) 15mA.
- Capacitive loading on ports 0 and 2 may cause the  $V_{OH}$  on ALE and PSEN to momentarily fall below the  $0.9V_{CC}$  specification when the address bits are stabilizing.
- Pins of ports 1, 2 and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when  $V_{IN}$  is approximately 2V.
- See Figures 10 through 13 for  $I_{CC}$  test conditions.
- The input threshold voltage of P1.6 and P1.7 (SI01) meets the I<sup>2</sup>C specification, so an input voltage below 1.5V will be recognized as a logic 0 while an input voltage above 3.0V will be recognized as a logic 1.

## CMOS single-chip 8-bit microcontroller 87C524

## AC ELECTRICAL CHARACTERISTICS

 $T_{amb} = 0^{\circ}\text{C to } +70^{\circ}\text{C or } -40^{\circ}\text{C to } +85^{\circ}\text{C}, V_{CC} = 5\text{V} \pm 10\%, V_{SS} = 0\text{V}^{1,2}$ 

SYMBOL	FIGURE	PARAMETER	16MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
$f_{t_{CLCL}}$	1	Oscillator frequency: Speed Versions 87C524 E 87C524 G			3.5 3.5	16 20	MHz MHz
$t_{HLH}$	1	ALE pulse width	85		$2t_{CLCL}-40$		ns
$t_{AVLL}$	1	Address valid to ALE low	8		$t_{CLCL}-55$		ns
$t_{LLAX}$	1	Address hold after ALE low	28		$t_{CLCL}-35$		ns
$t_{LLIV}$	1	ALE low to valid instruction in		150		$4t_{CLCL}-100$	ns
$t_{LLPL}$	1	ALE low to PSEN low	23		$t_{CLCL}-40$		ns
$t_{PLPH}$	1	PSEN pulse width	143		$3t_{CLCL}-45$		ns
$t_{PLIV}$	1	PSEN low to valid instruction in		83		$3t_{CLCL}-105$	ns
$t_{PXIX}$	1	Input instruction hold after PSEN	0		0		ns
$t_{PXIZ}$	1	Input instruction float after PSEN		38		$t_{CLCL}-25$	ns
$t_{AVIV}$	1	Address to valid instruction in		208		$5t_{CLCL}-105$	ns
$t_{PLAZ}$	1	PSEN low to address float		10		10	ns
<b>Data Memory</b>							
$t_{RLRH}$	2, 3	RD pulse width	275		$6t_{CLCL}-100$		ns
$t_{WLWH}$	2, 3	WR pulse width	275		$6t_{CLCL}-100$		ns
$t_{RLDV}$	2, 3	RD low to valid data in		148		$5t_{CLCL}-165$	ns
$t_{RHDZ}$	2, 3	Data hold after RD	0		0		ns
$t_{RHDZ}$	2, 3	Data float after RD		55		$2t_{CLCL}-70$	ns
$t_{LLDV}$	2, 3	ALE low to valid data in		350		$8t_{CLCL}-150$	ns
$t_{AVDV}$	2, 3	Address to valid data in		398		$9t_{CLCL}-165$	ns
$t_{LLWL}$	2, 3	ALE low to RD or WR low	138	238	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
$t_{AVWL}$	2, 3	Address valid to WR low or RD low	120		$4t_{CLCL}-130$		ns
$t_{QVWX}$	2, 3	Data valid to WR transition	3		$t_{CLCL}-60$		ns
$t_{WHQX}$	2, 3	Data hold after WR	13		$t_{CLCL}-50$		ns
$t_{RLAZ}$	2, 3	RD low to address float		0		0	ns
$t_{WHLH}$	2, 3	RD or WR high to ALE high	23	103	$t_{CLCL}-40$	$t_{CLCL}+40$	ns
<b>External Clock</b>							
$t_{CHCX}$	6	High time	20		20		ns
$t_{CLCX}$	6	Low time	20		20		ns
$t_{CLCH}$	6	Rise time		20		20	ns
$t_{CHCL}$	6	Fall time		20		20	ns
<b>Shift Register</b>							
$t_{XLXL}$	4	Serial port clock cycle time	750		$12t_{CLCL}$		ns
$t_{QVXH}$	4	Output data setup to clock rising edge	492		$10t_{CLCL}-133$		ns
$t_{XHQX}$	4	Output data hold after clock rising edge	8		$2t_{CLCL}-117$		ns
$t_{XHDZ}$	4	Input data hold after clock rising edge	0		0		ns
$t_{XHDV}$	4	Clock rising edge to input data valid		492		$10t_{CLCL}-133$	ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.

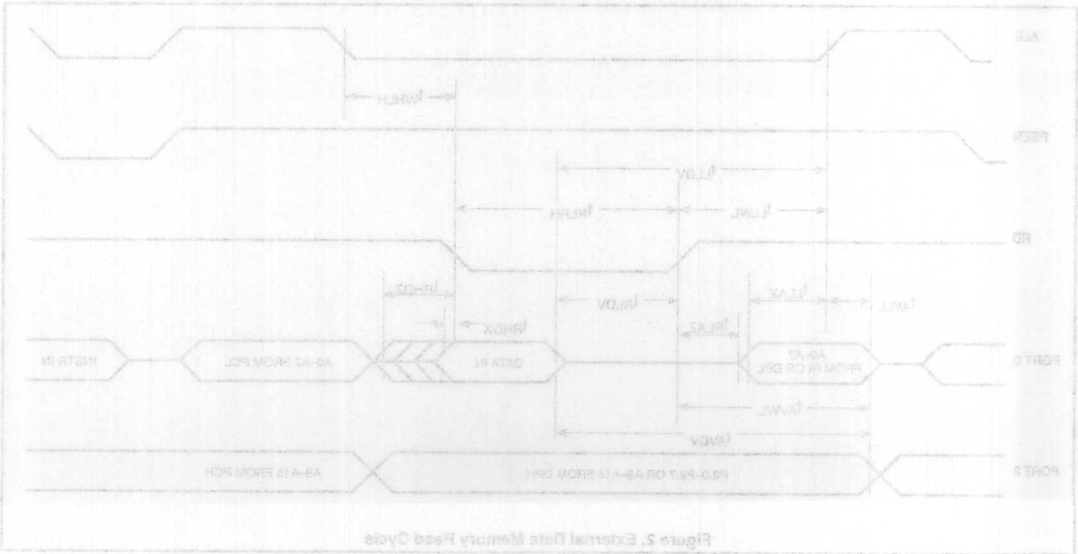
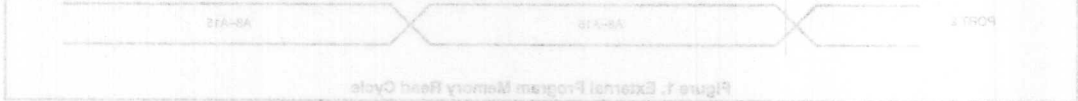


CMOS single-chip 8-bit microcontroller 87C524

AC ELECTRICAL CHARACTERISTICS – I<sup>2</sup>C INTERFACE

SYMBOL	PARAMETER	INPUT	OUTPUT	I <sup>2</sup> C SPECIFICATION
SCL Timing Characteristics				
t <sub>HD</sub> ; STA	START condition hold time	≥ 14 t <sub>CLCL</sub> <sup>1</sup>	Note 2	≥ 4.0μs
t <sub>LOW</sub>	SCL LOW time	≥ 16 t <sub>CLCL</sub>	Note 2	≥ 4.7μs
t <sub>HIGH</sub>	SCL HIGH time	≥ 14 t <sub>CLCL</sub> <sup>1</sup>	≥ 80 t <sub>CLCL</sub> <sup>3</sup>	≥ 4.0μs
t <sub>RC</sub>	SCL rise time	≤ 1μs <sup>4</sup>	Note 5	≤ 1.0μs
t <sub>FC</sub>	SCL fall time	≤ 0.3μs <sup>4</sup>	≤ 0.3μs <sup>6</sup>	≤ 0.3μs
SDA Timing Characteristics				
t <sub>SU</sub> ; DAT	Data set-up time	≥ 250ns	Note 2	≥ 250ns
t <sub>HD</sub> ; DAT	Data hold time	≥ 0ns	Note 2	≥ 0ns
t <sub>SU</sub> ; STA	Repeated START set-up time	≥ 14 t <sub>CLCL</sub> <sup>1</sup>	Note 2	≥ 4.7μs
t <sub>SU</sub> ; STO	STOP condition set-up time	≥ 14 t <sub>CLCL</sub> <sup>1</sup>	Note 2	≥ 4.0μs
t <sub>BUF</sub>	Bus free time	≥ 14 t <sub>CLCL</sub> <sup>1</sup>	Note 2	≥ 4.7μs
t <sub>RD</sub>	SDA rise time	≤ 1μs <sup>4</sup>	Note 5	≤ 1.0μs
t <sub>FD</sub>	SDA fall time	≤ 0.3μs <sup>4</sup>	≤ 0.3μs <sup>6</sup>	≤ 0.3μs

- NOTES:
- At f<sub>CLK</sub> = 3.5MHz, this evaluates to 14 × 286ns = 4μs, i.e., the bit-level I<sup>2</sup>C interface can respond to the I<sup>2</sup>C protocol for f<sub>CLK</sub> ≥ 3.5MHz.
  - This parameter is determined by the user software, it has to comply with the I<sup>2</sup>C specification.
  - This value gives the autclock pulse length which meets the I<sup>2</sup>C specification for the specified XTAL clock frequency range. Alternatively, the SCL pulse may be timed by software.
  - Spikes on SDA and SCL lines with a duration of less than 4 × f<sub>CLK</sub> will be filtered out.
  - The rise time is determined by the external bus line capacitance and pull-up resistor, it must be ≤ 1μs.
  - The maximum capacitance on bus lines SDA and SCL is 400pF.





## CMOS single-chip 8-bit microcontroller 87C524

## EXPLANATION OF THE AC SYMBOLS

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

A – Address

C - Clock

D – Input data

H – Logic level high

- Instruction (program memory contents)

L – Logic level low, or ALE

P - PSEN

Q – Output data

R -  $\overline{RD}$  signal

t - Time

Y – Valid

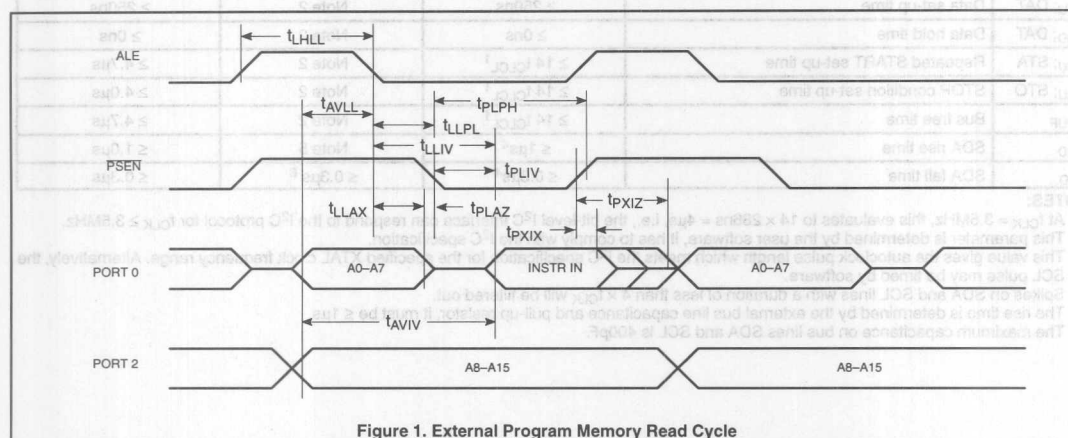
W-  $\overline{WR}$  signal

X – No longer a valid logic level

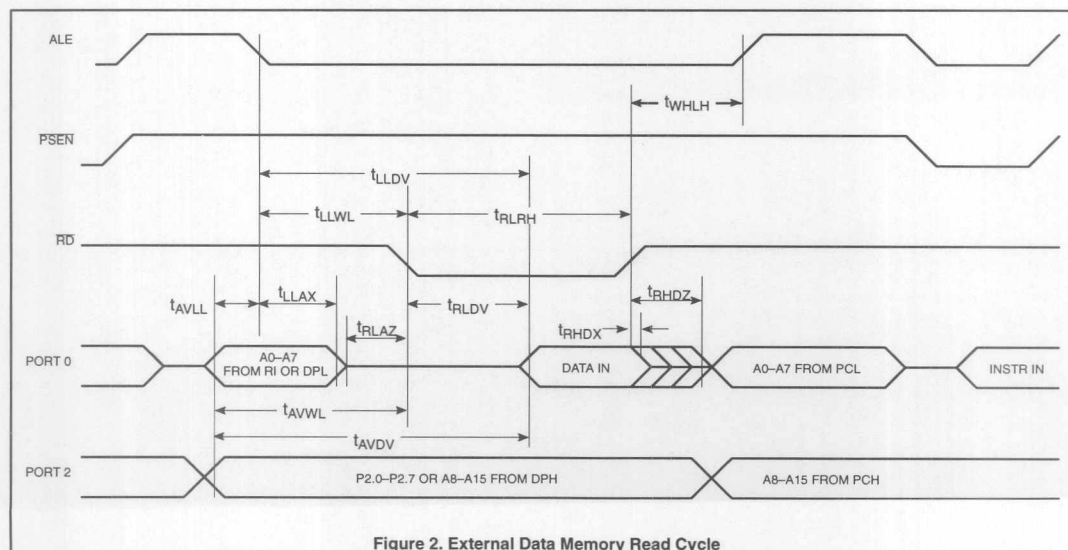
7 - Float

**Examples:**  $t_{AVLL}$  = Time for address valid to ALE low

$t_{LLPL}$  = Time for ALE low to  $\overline{PSEN}$  low.



**Figure 1. External Program Memory Read Cycle**



### Figure 2. External Data Memory Read Cycle

## CMOS single-chip 8-bit microcontroller

## 87C524

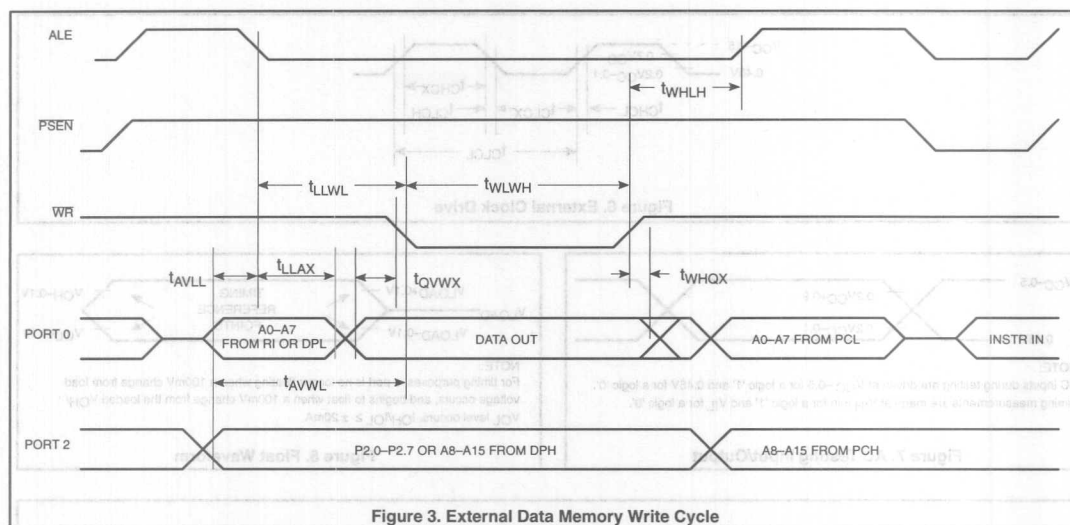


Figure 3. External Data Memory Write Cycle

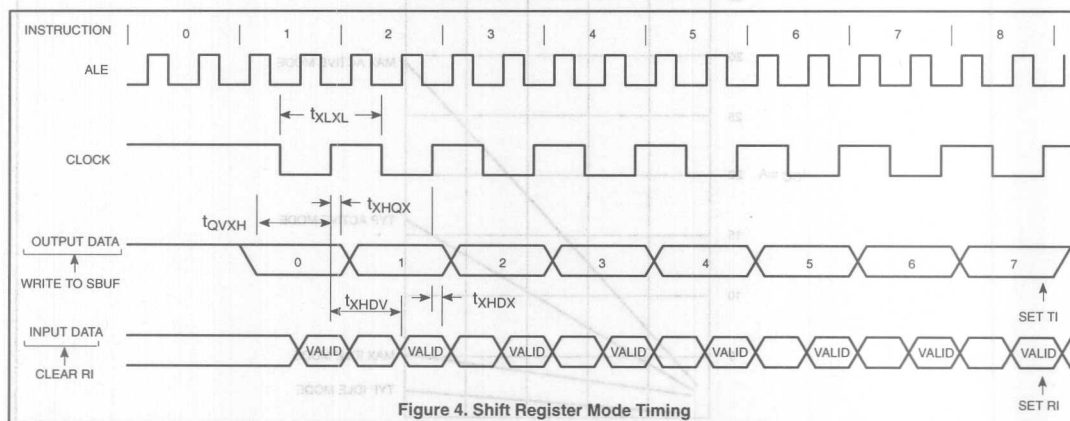
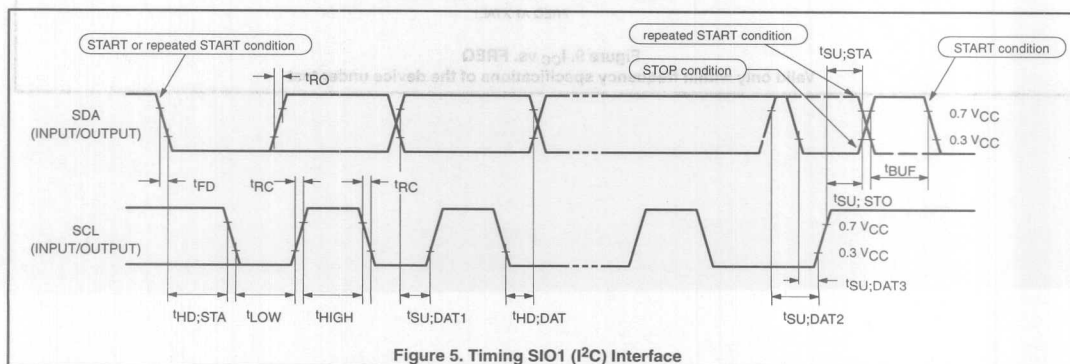


Figure 4. Shift Register Mode Timing

Figure 5. Timing SIO1 (I<sup>2</sup>C) Interface

## CMOS single-chip 8-bit microcontroller 87C524

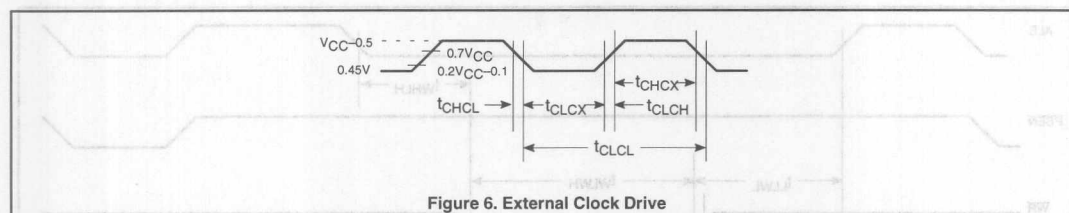


Figure 6. External Clock Drive

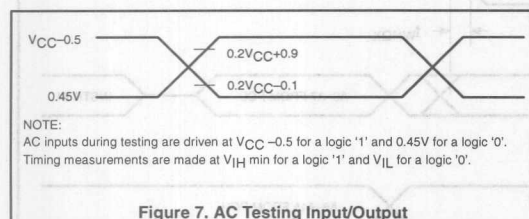


Figure 7. AC Testing Input/Output

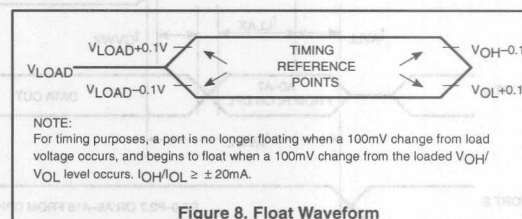
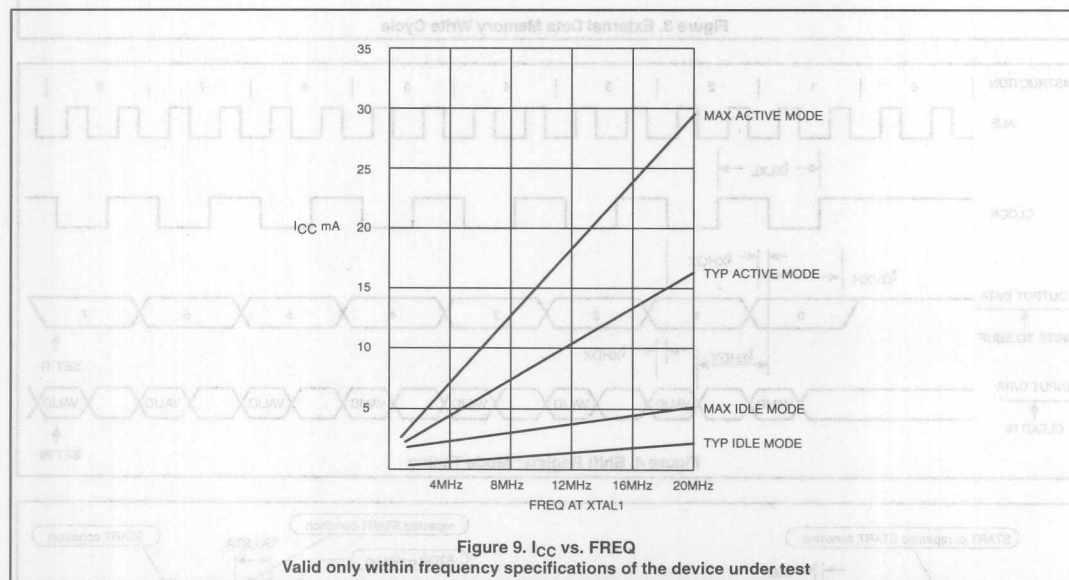
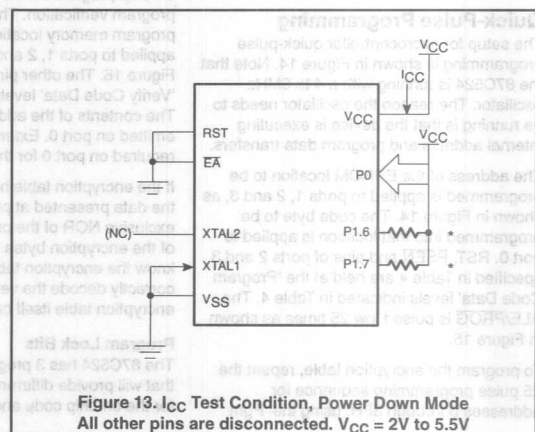
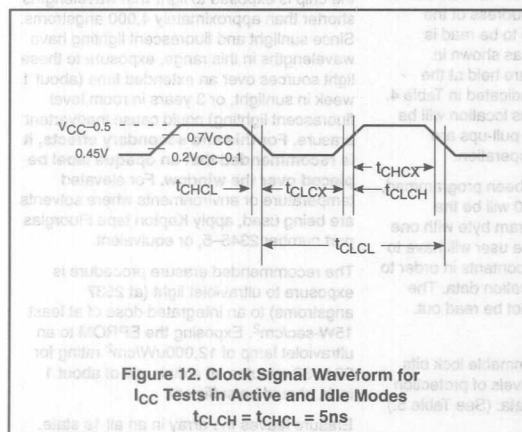
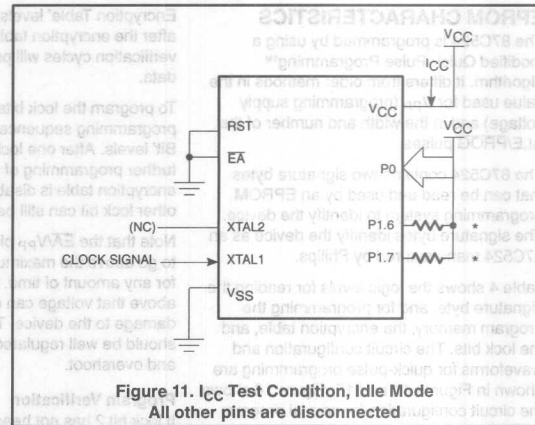
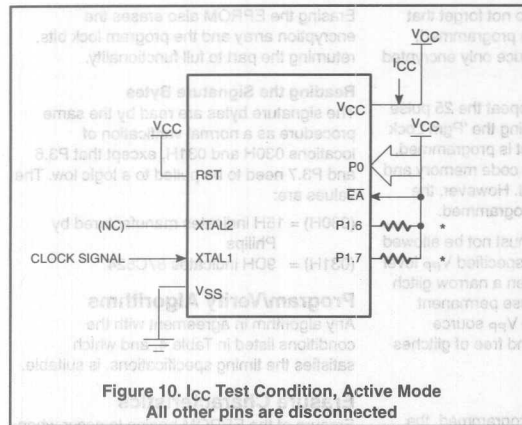


Figure 8. Float Waveform



## CMOS single-chip 8-bit microcontroller 87C524

**NOTE:**

- \* Ports 1.6 and 1.7 should be connected to  $V_{CC}$  through resistors of sufficiently high value such that the sink current into these pins does not exceed the  $I_{OL}$  specification.

## CMOS single-chip 8-bit microcontroller 87C524

**EPROM CHARACTERISTICS**

The 87C524 is programmed by using a modified Quick-Pulse Programming™ algorithm. It differs from older methods in the value used for  $V_{PP}$  (programming supply voltage) and in the width and number of the ALE/PROG pulses.

The 87C524 contains two signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes identify the device as an 87C524 manufactured by Philips.

Table 4 shows the logic levels for reading the signature byte, and for programming the program memory, the encryption table, and the lock bits. The circuit configuration and waveforms for quick-pulse programming are shown in Figures 14 and 15. Figure 16 shows the circuit configuration for normal program memory verification.

**Quick-Pulse Programming**

The setup for microcontroller quick-pulse programming is shown in Figure 14. Note that the 87C524 is running with a 4 to 6MHz oscillator. The reason the oscillator needs to be running is that the device is executing internal address and program data transfers.

The address of the EPROM location to be programmed is applied to ports 1, 2 and 3, as shown in Figure 14. The code byte to be programmed into that location is applied to port 0. RST, PSEN and pins of ports 2 and 3 specified in Table 4 are held at the 'Program Code Data' levels indicated in Table 4. The ALE/PROG is pulsed low 25 times as shown in Figure 15.

To program the encryption table, repeat the 25 pulse programming sequence for addresses 0 through 3FH, using the 'Pgm

Encryption Table' levels. Do not forget that after the encryption table is programmed, verification cycles will produce only encrypted data.

To program the lock bits, repeat the 25 pulse programming sequence using the 'Pgm Lock Bit' levels. After one lock bit is programmed, further programming of the code memory and encryption table is disabled. However, the other lock bit can still be programmed.

Note that the  $\overline{EA}/V_{PP}$  pin must not be allowed to go above the maximum specified  $V_{PP}$  level for any amount of time. Even a narrow glitch above that voltage can cause permanent damage to the device. The  $V_{PP}$  source should be well regulated and free of glitches and overshoot.

**Program Verification**

If lock bit 2 has not been programmed, the on-chip program memory can be read out for program verification. The address of the program memory locations to be read is applied to ports 1, 2 and 3 as shown in Figure 16. The other pins are held at the 'Verify Code Data' levels indicated in Table 4. The contents of the address location will be emitted on port 0. External pull-ups are required on port 0 for this operation.

If the encryption table has been programmed, the data presented at port 0 will be the exclusive NOR of the program byte with one of the encryption bytes. The user will have to know the encryption table contents in order to correctly decode the verification data. The encryption table itself cannot be read out.

**Program Lock Bits**

The 87C524 has 3 programmable lock bits that will provide different levels of protection for the on-chip code and data. (See Table 5.)

Erasing the EPROM also erases the encryption array and the program lock bits, returning the part to full functionality.

**Reading the Signature Bytes**

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. The values are:

(030H) = 15H indicates manufactured by Philips

(031H) = 9DH indicates 87C524

**Program/Verify Algorithms**

Any algorithm in agreement with the conditions listed in Table 4, and which satisfies the timing specifications, is suitable.

**Erasure Characteristics**

Erasure of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent erasure. **For this and secondary effects, it is recommended that an opaque label be placed over the window.** For elevated temperature or environments where solvents are being used, apply Kapton tape Fluorglas part number 2345-5, or equivalent.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 angstroms) to an integrated dose of at least 15W-sec/cm<sup>2</sup>. Exposing the EPROM to an ultraviolet lamp of 12,000uW/cm<sup>2</sup> rating for 20 to 39 minutes, at a distance of about 1 inch, should be sufficient.

Erasure leaves the array in an all 1s state.

™Trademark phrase of Intel Corporation.



## CMOS single-chip 8-bit microcontroller 87C524

Table 4. EPROM Programming Modes

MODE	RST	PSEN	ALE/PROG	EA/V <sub>PP</sub>	P2.7	P2.6	P3.7	P3.6
Read signature	1	0	1	1	0	0	0	0
Program code data	1	0	0*	V <sub>PP</sub>	1	0	1	1
Verify code data	1	0	1	1	0	0	1	1
Pgm encryption table	1	0	0*	V <sub>PP</sub>	1	0	1	0
Pgm lock bit 1	1	0	0*	V <sub>PP</sub>	1	1	1	1
Pgm lock bit 2	1	0	0*	V <sub>PP</sub>	1	1	0	0
Pgm lock bit 3	1	0	0*	V <sub>PP</sub>	0	1	0	1

## NOTES:

1. '0' = Valid low for that pin, '1' = valid high for that pin.

2. V<sub>PP</sub> = 12.75V ±0.25V.3. V<sub>CC</sub> = 5V±10% during programming and verification.\* ALE/PROG receives 25 programming pulses while V<sub>PP</sub> is held at 12.75V. Each programming pulse is low for 100µs (±10µs) and high for a minimum of 10µs.

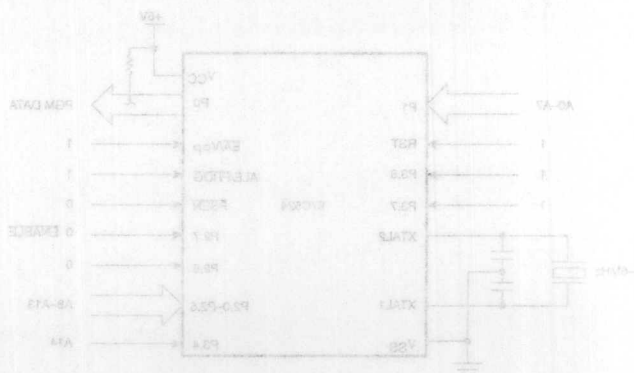
Table 5. Program Lock Bits

PROGRAM LOCK BITS <sup>1, 2</sup>				PROTECTION DESCRIPTION
	LB1	LB2	LB3	
1	U	U	U	No Program Lock features enabled. (Code verify will still be encrypted by the Encryption Array if programmed.)
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, EA is sampled and latched on Reset, and further programming of the EPROM is disabled.
3	P	P	U	Same as 2, also verify is disabled.
4	P	P	P	Same as 3, external execution is disabled. Internal data RAM is not accessible.

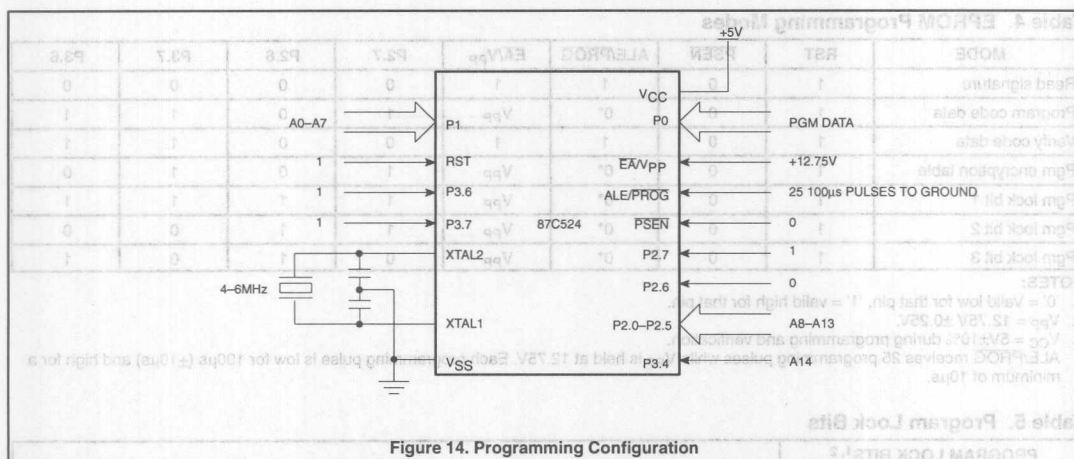
## NOTES:

1. P – programmed. U – unprogrammed.

2. Any other combination of the lock bits is not defined.



## CMOS single-chip 8-bit microcontroller 87C524



**Figure 14. Programming Configuration**

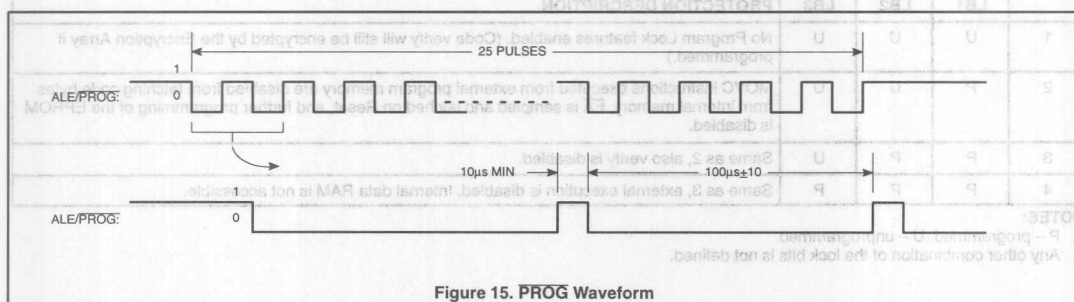


Figure 15.  $\overline{\text{PROG}}$  Waveform

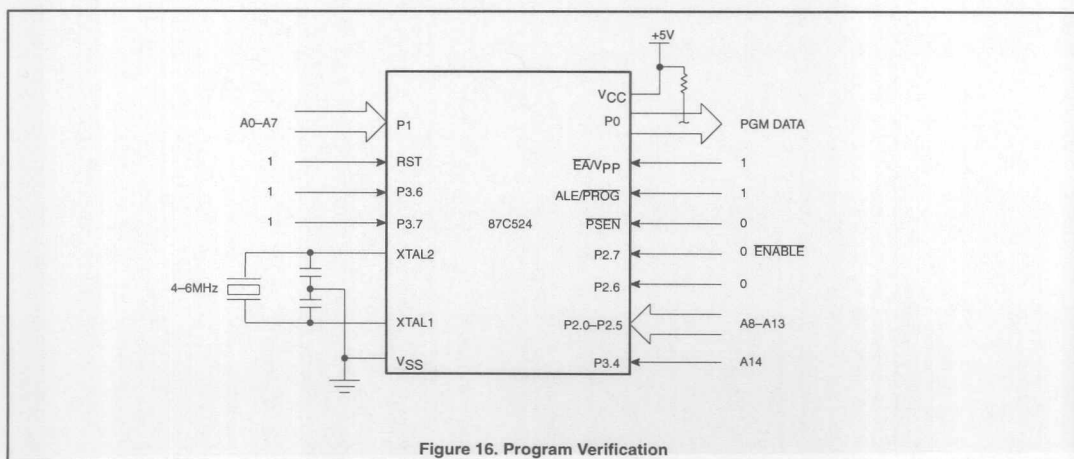


Figure 16. Program Verification

## CMOS single-chip 8-bit microcontroller 87C524

## EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

T<sub>amb</sub> = 21°C to +27°C, V<sub>CC</sub> = 5V±10%, V<sub>SS</sub> = 0V (See Figure 17)

SYMBOL	PARAMETER	MIN	MAX	UNIT
V <sub>PP</sub>	Programming supply voltage	12.5	13.0	V
I <sub>PP</sub>	Programming supply current		50	mA
1/t <sub>CLCL</sub>	Oscillator frequency	4	6	MHz
t <sub>AVGL</sub>	Address setup to PROG low	48t <sub>CLCL</sub>		ns
t <sub>GHAX</sub>	Address hold after PROG	48t <sub>CLCL</sub>		ns
t <sub>DVGL</sub>	Data setup to PROG low	48t <sub>CLCL</sub>		ns
t <sub>GHDX</sub>	Data hold after PROG	48t <sub>CLCL</sub>		ns
t <sub>EHSH</sub>	P2.7 (ENABLE) high to V <sub>PP</sub>	48t <sub>CLCL</sub>		ns
t <sub>SHGL</sub>	V <sub>PP</sub> setup to PROG low	10		µs
t <sub>GHSL</sub>	V <sub>PP</sub> hold after PROG	10		µs
t <sub>GLGH</sub>	PROG width	90	110	µs
t <sub>AVQV</sub>	Address to data valid		48t <sub>CLCL</sub>	ns
t <sub>ELQZ</sub>	ENABLE low to data valid		48t <sub>CLCL</sub>	ns
t <sub>EHQZ</sub>	Data float after ENABLE	0	48t <sub>CLCL</sub>	ns
t <sub>GHGL</sub>	PROG high to PROG low	10		µs

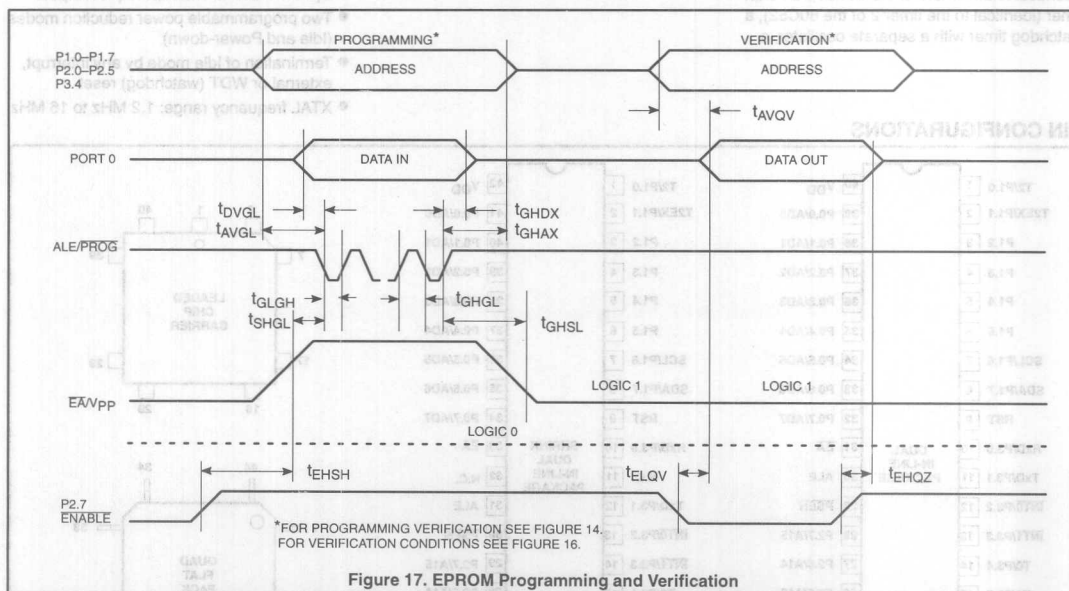


Figure 17. EPROM Programming and Verification



Purchase of Philips' I<sup>2</sup>C components conveys a license under the Philips' I<sup>2</sup>C patent to use the components in the I<sup>2</sup>C-system provided the system conforms to the I<sup>2</sup>C specifications defined by Philips.

# CMOS single-chip 8-bit microcontroller 80C528/83C528

## DESCRIPTION

The 8XC528 single-chip 8-bit microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 8XC528 has the same instruction set as the 80C51. Three versions of the derivative exist:

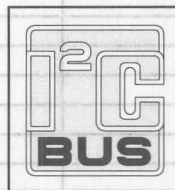
- 83C528 — 32k bytes mask programmable ROM
- 80C528 — ROMless version of the 83C528
- 87C528 — 32k bytes EPROM (described in a separate data sheet)

This device provides architectural enhancements that make it applicable in a variety of applications in consumer, telecom and general control systems, especially in those systems which need large ROM and RAM capacity on-chip.

The 8XC528 contains a 32k × 8 ROM (83C528), a 512 × 8 RAM, four 8-bit I/O ports, two 16-bit timer/event counters (identical to the timers of the 80C51), a 16-bit timer (identical to the timer 2 of the 80C52), a watchdog timer with a separate oscillator, a

multi-source, two-priority-level, nested interrupt structure, two serial interfaces (UART and I<sup>2</sup>C-bus), and on-chip oscillator and timing circuits.

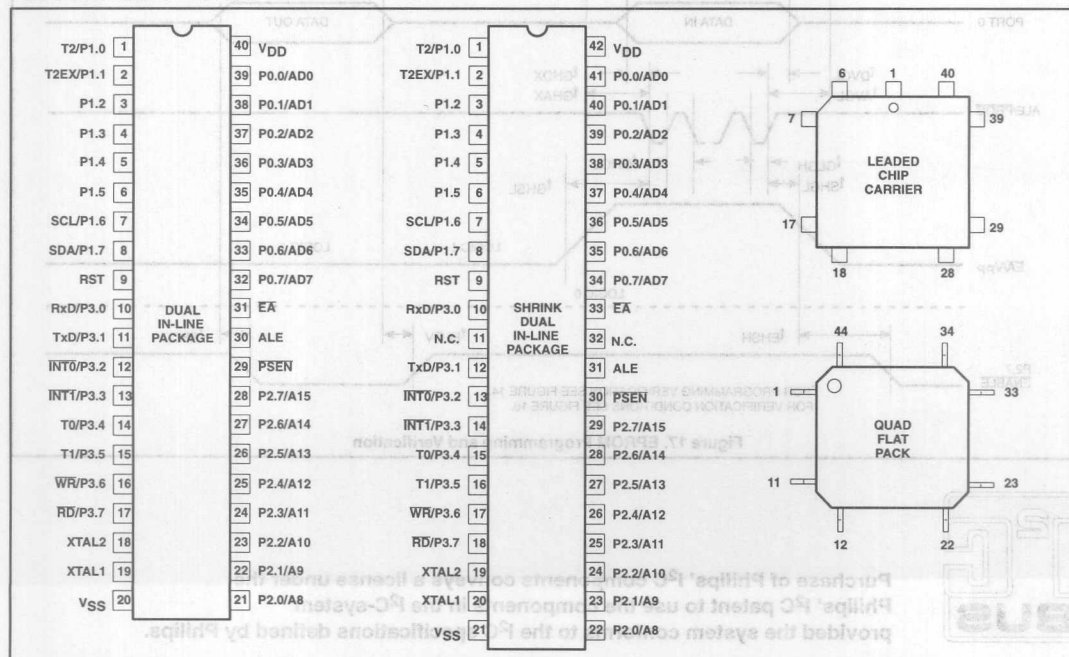
In addition, the 8XC528 has two software selectable modes of power reduction — idle mode and power-down mode. The idle mode freezes the CPU while allowing the RAM, timers, serial port, and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.



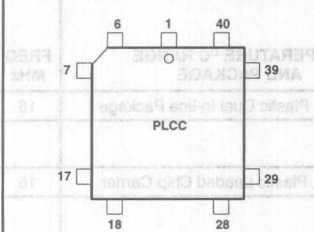
## FEATURES

- 80C51 instruction set
- 32k × 8 ROM (83C528)
- ROMless (80C528)
- 512 × 8 RAM
- Memory addressing capability 64k ROM and 64k RAM
- Three 16-bit counter/timers
- On-chip watchdog timer with oscillator
- Full duplex UART
- I<sup>2</sup>C serial interface
- Four 8-bit I/O ports
- Power control modes:
  - Idle mode
  - Power-down mode
  - Warm start from power-down
- CMOS and TTL compatible
- Extended temperature ranges
- ROM code protection
- 7-source and 7-vector interrupt structure with 2 priority levels
- Up to 3 external interrupt request inputs
- Two programmable power reduction modes (Idle and Power-down)
- Termination of Idle mode by any interrupt, external or WDT (watchdog) reset
- XTAL frequency range: 1.2 MHz to 16 MHz

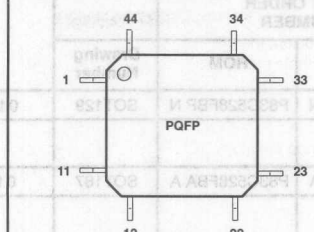
## PIN CONFIGURATIONS



## CMOS single-chip 8-bit microcontroller 80C528/83C528

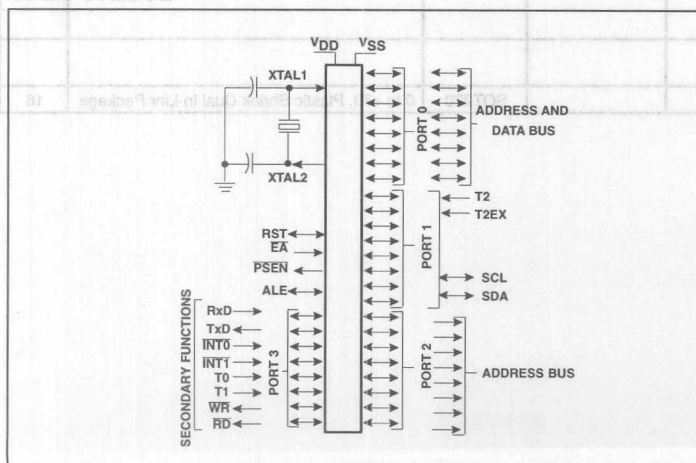
CERAMIC AND PLASTIC LEADED  
CHIP CARRIER PIN FUNCTIONS


Pin	Function	Pin	Function
1	NC	23	NC
2	P1.0/T2	24	P2.0/A8
3	P1.1/T2EX	25	P2.1/A9
4	P1.2	26	P2.2/A10
5	P1.3	27	P2.3/A11
6	P1.4	28	P2.4/A12
7	P1.5	29	P2.5/A13
8	P1.6/SCL	30	P2.6/A14
9	P1.7/SDA	31	P2.7/A15
10	RST	32	PSEN
11	P3.0/RxD	33	ALE
12	NC	34	NC
13	P3.1/TxD	35	EA
14	P3.2/INT0	36	P0.7/AD7
15	P3.3/INT1	37	P0.6/AD6
16	P3.4/T0	38	P0.5/AD5
17	P3.5/T1	39	P0.4/AD4
18	P3.6/WR	40	P0.3/AD3
19	P3.7/RD	41	P0.2/AD2
20	XTAL2	42	P0.1/AD1
21	XTAL1	43	P0.0/AD0
22	VSS	44	VDD

PLASTIC QUAD FLAT PACK  
PIN FUNCTIONS


Pin	Function	Pin	Function
1	P1.5	23	P2.5/A13
2	P1.6/SCL	24	P2.6/A14
3	P1.7/SDA	25	P2.7/A15
4	RST	26	PSEN
5	P3.0/RxD	27	ALE
6	NC	28	NC
7	P3.1/TxD	29	EA
8	P3.2/INT0	30	P0.7/AD7
9	P3.3/INT1	31	P0.6/AD6
10	P3.4/T0	32	P0.5/AD5
11	P3.5/T1	33	P0.4/AD4
12	P3.6/WR	34	P0.3/AD3
13	P3.7/RD	35	P0.2/AD2
14	XTAL2	36	P0.1/AD1
15	XTAL1	37	P0.0/AD0
16	VSS	38	VDD
17	NC	39	NC
18	P2.0/A8	40	P1.0/T2
19	P2.1/A9	41	P1.1/T2EX
20	P2.2/A10	42	P1.2
21	P2.3/A11	43	P1.3
22	P2.4/A12	44	P1.4

## LOGIC SYMBOL





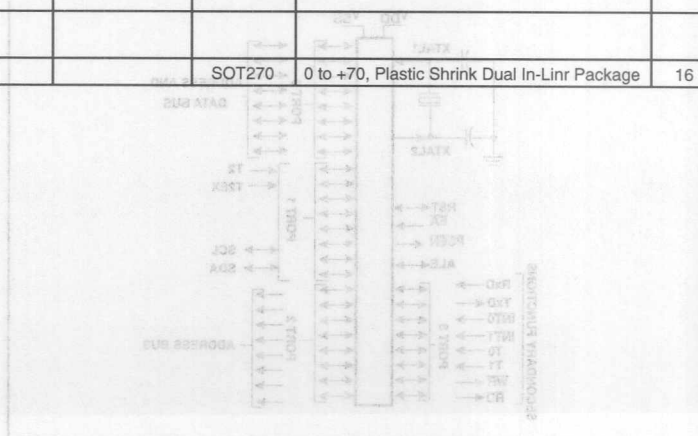
## CMOS single-chip 8-bit microcontroller 80C528/83C528

## ORDERING INFORMATION

[illegible]

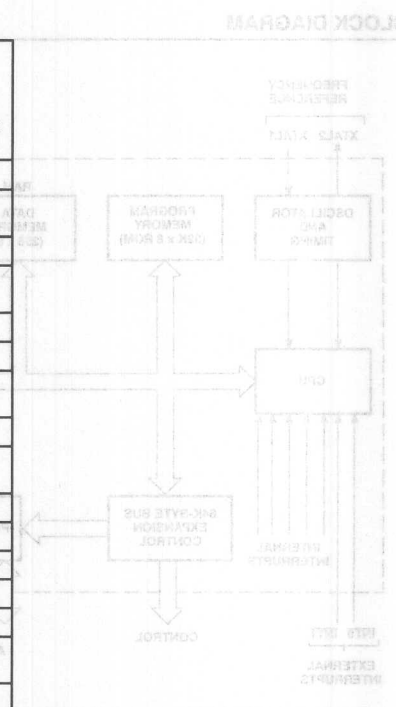
**NOTE:**

1. xxx denotes the ROM code number.
2. SOT311 replaced by SOT307-2.



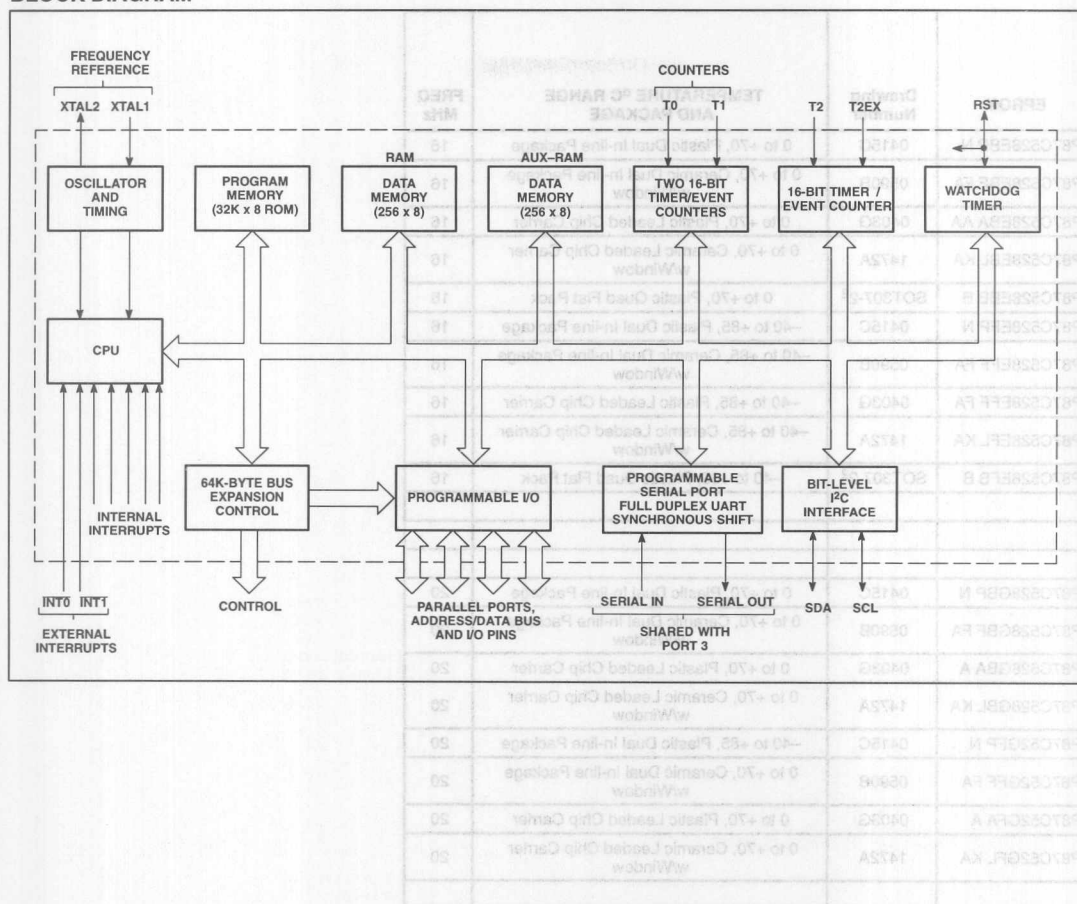
## CMOS single-chip 8-bit microcontroller 80C528/83C528

EPROM	Drawing Number	TEMPERATURE °C RANGE AND PACKAGE	FREQ MHz
P87C528EBP N	0415C	0 to +70, Plastic Dual In-line Package	16
P87C528EBF FA	0590B	0 to +70, Ceramic Dual In-line Package w/Window	16
P87C528EBA AA	0403G	0 to +70, Plastic Leaded Chip Carrier	16
P87C528EBL KA	1472A	0 to +70, Ceramic Leaded Chip Carrier w/Window	16
P87C528EBB B	SOT307-2 <sup>2</sup>	0 to +70, Plastic Quad Flat Pack	16
P87C528EFP N	0415C	-40 to +85, Plastic Dual In-line Package	16
P87C528EFF FA	0590B	-40 to +85, Ceramic Dual In-line Package w/Window	16
P87C528EFF FA	0403G	-40 to +85, Plastic Leaded Chip Carrier	16
P87C528EFL KA	1472A	-40 to +85, Ceramic Leaded Chip Carrier w/Window	16
P87C528EFB B	SOT307-2 <sup>2</sup>	-40 to +85, Plastic Quad Flat Pack	16
P87C528GBP N	0415C	0 to +70, Plastic Dual In-line Package	20
P87C528GBF FA	0590B	0 to +70, Ceramic Dual In-line Package w/Window	20
P87C528GBA A	0403G	0 to +70, Plastic Leaded Chip Carrier	20
P87C528GBL KA	1472A	0 to +70, Ceramic Leaded Chip Carrier w/Window	20
P87C528GFP N	0415C	-40 to +85, Plastic Dual In-line Package	20
P87C528GFF FA	0590B	0 to +70, Ceramic Dual In-line Package w/Window	20
P87C528GFA A	0403G	0 to +70, Plastic Leaded Chip Carrier	20
P87C528GFL KA	1472A	0 to +70, Ceramic Leaded Chip Carrier w/Window	20



## CMOS single-chip 8-bit microcontroller 80C528/83C528

## BLOCK DIAGRAM



## CMOS single-chip 8-bit microcontroller 80C528/83C528

## PIN DESCRIPTION

MNEMONIC	PIN NO.				TYPE	NAME AND FUNCTION
	DIP	SDIL	LCC	QFP		
V <sub>SS</sub>	20	21	22	16	I	<b>Ground:</b> circuit ground potential.
V <sub>DD</sub>	40	42	44	38	I	
P0.0–0.7	39–32	41–34	43–36	37–30	I/O	<b>Port 0:</b> Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s.
P1.0–P1.7	1–8	1–8	2–9	40–44	I/O	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pull-ups, except P1.6 and P1.7 which have open drain. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 1 can sink/source one TTL (4 LSTTL) inputs.
P2.0–P2.7	1	1	2	40	I	<b>T2 (P1.0):</b> Timer/counter 2 external count input (following edge triggered).
	2	2	3	41	I	<b>T2EX (P1.1):</b> Timer/counter 2 trigger input.
	7	7	8	2	I/O	<b>SCL (P1.6):</b> I <sup>2</sup> C serial port clock line.
	8	8	9	3	I/O	<b>SDA (P1.7):</b> I <sup>2</sup> C serial port data line.
P2.0–P2.7	21–28	22–29	24–31	18–25	I/O	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register.
P3.0–P3.7	10–17	10–18 (11=NC)	11, 13–19	5, 7–13	I/O	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 3 also serves the special features of the SC80C51 family, as listed below:
RST	10	10	11	5	I	<b>RxD (P3.0):</b> Serial input port
	11	12	13	7	O	<b>TxD (P3.1):</b> Serial output port
	12	13	14	8	I	<b>INT0 (P3.2):</b> External interrupt
	13	14	15	9	I	<b>INT1 (P3.3):</b> External interrupt
	14	15	16	10	I	<b>T0 (P3.4):</b> Timer 0 external input
	15	16	17	11	I	<b>T1 (P3.5):</b> Timer 1 external input
	16	17	18	12	O	<b>WR (P3.6):</b> External data memory write strobe
	17	18	19	13	O	<b>RD (P3.7):</b> External data memory read strobe
RST	9	9	10	4	I/O	<b>Reset:</b> A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to V <sub>SS</sub> permits a power-on reset using only an external capacitor to V <sub>DD</sub> . After a watchdog timer overflow, this pin is pulled high while the internal reset signal is active.
ALE	30	31	33	27	I/O	<b>Address Latch Enable:</b> Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory.
PSEN	29	30	32	26	O	<b>Program Store Enable:</b> The read strobe to external program memory. When the device is executing code from the external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory. PSEN is not activated during fetches from internal program memory.
E <sub>A</sub>	31	33	35	29	I	<b>External Access Enable:</b> E <sub>A</sub> must be externally held low during RESET to enable the device to fetch code from external program memory locations 0000H to 7FFFH. If E <sub>A</sub> is held high during RESET, the device executes from internal program memory unless the program counter contains an address greater than 7FFFH. E <sub>A</sub> is don't care after RESET.
XTAL1	19	20	21	15	I	<b>Crystal 1:</b> Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	19	20	14	O	<b>Crystal 2:</b> Output from the inverting oscillator amplifier.

## CMOS single-chip 8-bit microcontroller 80C528/83C528

Table 1. 8XC524/8XC528 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB							LSB	
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
DPTR:	Data pointer (2 bytes):										
DPH	Data pointer high	83H									00H
DPL	Data pointer low	82H									00H
IE*#	Interrupt enable	A8H	AF	AE	AD	AC	AB	AA	A9	A8	
			EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0	00H
			BF	BE	BD	BC	BB	BA	B9	B8	
IP*#	Interrupt priority	B8H	—	PS1	PT2	PS0	PT1	PX1	PT0	PX0	x0000000B
			87	86	85	84	83	82	81	80	
P0*	Port 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
			97	96	95	94	93	92	91	90	
P1*	Port 1	90H	SDA	SEL	—	—	—	—	T2EX	T2	FFH
			A7	A6	A5	A4	A3	A2	A1	A0	
P2*	Port 2	A0H	A15	A14	A13	A12	A11	A10	A9	A8	FFH
			B7	B6	B5	B4	B3	B2	B1	B0	
P3*	Port 3	B0H	RD	WR	T1	T0	INT1	INT0	TxD	RxD	FFH
PCON	Power control	87H	SMOD	—	—	—	GF1	GF0	PD	IDL	0xxx0000B
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program status word	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	00H
RCAP2H#	Capture high	CBH									00H
RCAP2L#	Capture low	CAH									00H
SBUF	Serial data buffer	99H									xxxxxxxB
			9F	9E	9D	9C	9B	9A	99	98	
SCON*	Serial controller	98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00H
S1BIT#	Serial I <sup>2</sup> C data	D9H/RD	SDI	0	0	0	0	0	0	0	x0000000B
			SD0	X	X	X	X	X	X	X	0xxxxxxxB
S1INT#	Serial I <sup>2</sup> C interrupt	DAH	INT	X	X	X	X	X	X	X	0xxxxxxxB
			DF	DE	DD	DC	DB	DA	D9	D8	
S1SCS*#	Serial I <sup>2</sup> C control	D8H/RD	SDI	SCI	CLH	BB	RBF	WBF	STR	ENS	xxxx0000B
		WR	SD0	SC0	CLH	X	X	X	STR	ENS	00xxxx00B
SP	Stack pointer	81H									07H
			8F	8E	8D	8C	8B	8A	89	88	
TCON*	Timer control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
			CF	CE	CD	CC	CB	CA	C9	C8	
T2CON*#	Timer 2 control	C8H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	00H
TH0	Timer high 0	8CH									00H
TH1	Timer high 1	8DH									00H
TH2#	Timer high 2	CDH									00H
TL0	Timer low 0	8AH									00H
TL1	Timer low 1	8BH									00H
TL2#	Timer low 2	CCH									00H
T3#	Watchdog timer	FFH									00H
TMOD	Timer mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H
WDCON#	Watchdog control	A5H									A5H

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.



## CMOS single-chip 8-bit microcontroller 80C528/83C528

Table 2. Internal and External Program Memory Access with Security Bit Set

INSTRUCTION	ACCESS TO INTERNAL PROGRAM MEMORY	ACCESS TO EXTERNAL PROGRAM MEMORY
MOVC in internal program memory	YES	YES
MOVC in external program memory	NO	YES

## ROM CODE PROTECTION

By setting a mask programmable security bit, the ROM content in the 83C528 is protected, i.e., it cannot be read out by any test mode or by any instruction in the external program memory space. The MOVC instructions are the only ones which have access to program code in the internal or external program memory. The EA input is latched during RESET and is 'don't care' after RESET (also if security bit is not set). This implementation prevents reading from internal program code by switching from external program memory to internal program memory during MOVC instruction or an instruction that handles immediate data. Table 2 lists the access to the internal and external program memory by the MOVC instructions when the security bit has been set to logical one. If the security bit has been set to a logical 0 there are no restrictions for the MOVC instructions.

## INTERNAL DATA MEMORY

The internal data memory is divided into three physically separated segments: 256 bytes of RAM, 256 bytes of AUX-RAM, and a 128 bytes special function area. These can be addressed each in a different way.

- RAM 0 to 127 can be addressed directly and indirectly as in the 80C51. Address pointers are R0 and R1 of the selected register bank.
- RAM 128 to 255 can only be addressed indirectly as in the 80C51. Address pointers are R0 and R1 of the selected register bank.
- AUX-RAM 0 to 255 is indirectly addressed in the same way as external data memory with the MOVX instructions. Address pointers are R0, R1 of the selected register bank and DPTR. An access to AUX-RAM 0 to 255 will not affect ports P0, P2, P3.6 and P3.7.

An access to external data memory locations higher than 255 will be performed with the MOVX DPTR instructions in the same way as in the 8051 structure, so with P0 and P2 as data/address bus and P3.6 and P3.7 as write and read timing signals. Note that these external data memory cannot be accessed with R0 and R1 as address pointer.

## TIMER 2

Timer 2 is functionally equal to the Timer 2 of the 8052AH. Timer 2 is a 16-bit timer/counter. These 16 bits are formed by two special function registers TL2 and TH2. Another pair of special function register RCAP2L and RCAP2H form a 16-bit capture register or a 16-bit reload register. Like Timer 0 and 1, it can operate either as a timer or as an event counter. This is selected by bit C/T2N in the special function register T2CON. It has three operating modes: capture, autoloader, and baud rate generator mode which are selected by bits in T2CON.

## WATCHDOG TIMER T3

The watchdog timer consists of an 11-bit prescaler and an 8-bit timer formed by special function register T3. The prescaler is incremented by an on-chip oscillator with a fixed frequency of 1MHz. The maximum tolerance on this frequency is -50% and +100%. The 8-bit timer increments every 2048 cycles of the on-chip oscillator. When a timer overflow occurs, the microcontroller is reset and a reset output pulse of  $16 \times 2048$  cycles of the on-chip oscillator is generated at pin RST. The internal RESET signal is not inhibited when the external RST pin is kept low by, for example, an external reset circuit. The RESET signal drives port 1, 2, 3 into the high state and port 0 into the high impedance state.

The watchdog timer is controlled by one special function register WDCON with the direct address location A5H. WDCON can be read and written by software. A value of A5H in WDCON halts the on-chip oscillator and clears both the prescaler and timer T3. After the RESET signal, WDCON contains A5H. Every value other than A5H in WDCON enables the watchdog timer. When the watchdog timer is enabled, it runs independently of the XTAL-clock.

Timer T3 can be read on the fly. Timer T3 can only be written if WDCON contains the value 5AH. A successful write operation to T3 will clear the prescaler and WDCON, leaving the watchdog enabled and preventing inadvertent changes of T3. To prevent an overflow of the watchdog timer, the user

program has to reload the watchdog timer within periods that are shorter than the programmed watchdog timer interval. This time interval is determined by an 8-bit value that has to be loaded in register T3 while at the same time the prescaler is cleared by hardware.

Watchdog timer interval =

$$\frac{[256 - (T3)] \times 2048}{\text{on-chip oscillator frequency}}$$

BIT-LEVEL I<sup>2</sup>C INTERFACE

This bit-level serial I/O interface supports the I<sup>2</sup>C-bus. P1.6/SCL and P1.7/SDA are the serial I/O pins. These two pins meet the I<sup>2</sup>C specification concerning the input levels and output drive capability. Consequently, these pins have an open drain output configuration. All the four modes of the I<sup>2</sup>C-bus are supported:

- master transmitter
- master receiver
- slave transmitter
- slave receiver

The advantages of the bit-level I<sup>2</sup>C hardware compared with a full software I<sup>2</sup>C implementation are:

- the hardware can generate the SCL pulse
- Testing a single bit (RBF respectively, WBF) is sufficient as a check for error free transmission.

The bit-level I<sup>2</sup>C hardware operates on serial bit level and performs the following functions:

- filtering the incoming serial data and clock signals
- recognizing the START condition
- generating a serial interrupt request SI after reception of a START condition and the first falling edge of the serial clock
- recognizing the STOP condition
- recognizing a serial clock pulse on the SCL line
- latching a serial bit on the SDA line (SDI)
- stretching the SCL LOW period of the serial clock to suspend the transfer of the next serial data bit
- setting Read Bit Finished (RBF) when the SCL clock pulse has finished and Write Bit

- Finished (WBF) if there is no arbitration loss detected (i.e., SDA = 0 while SDO = 1)
- setting a serial clock Low-to-High detected (CLH) flag
- setting a Bus Busy (BB) flag on a START condition and clearing this flag on a STOP condition
- releasing the SCL line and clearing the CLH, RBF and WBF flags to resume transfer of the next serial data bit
- generating an automatic clock if the single bit data register S1BIT is used in master mode.

The following functions must be done in software:

- handling the I<sup>2</sup>C START interrupts
- converting serial to parallel data when receiving
- converting parallel to serial data when transmitting
- comparing the received slave address with its own
- interpreting the acknowledge information
- guarding the I<sup>2</sup>C status if RBF or WBF = 0.

Additionally, if acting as master:

- generating START and STOP conditions
- handling bus arbitration
- generating serial clock pulses if S1BIT is not used.

Three SFRs control the bit-level I<sup>2</sup>C interface: S1INT, S1BIT and S1SCS.

## OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator, as shown in the Logic Symbol.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2

is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

## RESET

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. To insure a good power-up reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few milliseconds) plus two machine cycles. At power-up, the voltage on V<sub>DD</sub> and RST must come up at the same time for a proper start-up.

## IDLE MODE

In idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

## POWER-DOWN MODE

In the power-down mode, the oscillator is stopped and the instruction to invoke power-down is the last instruction executed. The power-down mode can be terminated by

a RESET in the same way as in the 80C51 or in addition by one of two external interrupts, INT0 or INT1. A termination with an external interrupt does not affect the internal data memory and does not affect the special function registers. This makes it possible to exit power-down without changing the port output levels. To terminate the power-down mode with an external interrupt INT0 or INT1 must be switched to level-sensitive and must be enabled. The external interrupt input signal INT0 and INT1 must be kept low until the oscillator has restarted and stabilized. An instruction following the instruction that puts the device in the power-down mode will be executed. A reset generated by the watchdog timer terminates the power-down mode in the same way as an external RESET, and only the contents of the on-chip RAM are preserved. The control bits for the reduced power modes are in the special function register PCON.

## DESIGN CONSIDERATIONS

At power-on, the voltage on V<sub>DD</sub> and RST must come up at the same time for a proper start-up.

When the idle mode is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write when idle is terminated by reset, the instruction following the one that invokes idle should not be one that writes to a port pin or to external memory.

Table 3 shows the state of I/O ports during low current operating modes.

**Table 3. External Pin Status During Idle and Power-Down Modes**

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

## CMOS single-chip 8-bit microcontroller 80C528/83C528

ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>

PARAMETER	RATING	UNIT
Operating temperature under bias	0 to +70, or -40 to +85, or -40 to +125	°C
Storage temperature range	-65 to +150	°C
Voltage on any other pin to V <sub>SS</sub>	-0.5 to V <sub>DD</sub> +0.5	V
Input, output current on any two pins	±10	mA
Power dissipation (based on package heat transfer limitations, not device power consumption)	1.0	W

## NOTES:

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
- This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
- Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V<sub>SS</sub> unless otherwise noted.

## DC ELECTRICAL CHARACTERISTICS

T<sub>amb</sub> = 0°C to +70°C (V<sub>DD</sub> = 5V ±20%), -40°C to +85°C (V<sub>DD</sub> = 5V ±20%), or -40°C to +125°C (V<sub>DD</sub> = 5V ±10%), V<sub>SS</sub> = 0V

SYMBOL	PARAMETER	PART TYPE	TEST CONDITIONS	LIMITS		UNIT
				MIN	MAX	
V <sub>IL</sub>	Input low voltage, except EA, P1.6/SCL, P1.7/SDA	0°C to 70°C		-0.5	0.2V <sub>DD</sub> -0.1	V
		-40°C to +85°C		-0.5	0.2V <sub>DD</sub> -0.15	V
		-40°C to +125°C		-0.5	0.2V <sub>DD</sub> -0.25	V
V <sub>IL1</sub>	Input low voltage to EA	0°C to 70°C		-0.5	0.2V <sub>DD</sub> -0.3	V
		-40°C to +85°C		-0.5	0.2V <sub>DD</sub> -0.35	V
		-40°C to +125°C		-0.5	0.2V <sub>DD</sub> -0.45	V
V <sub>IL2</sub>	Input low voltage to P1.6/SCL, P1.7/SDA <sup>3</sup>			-0.5	0.3V <sub>DD</sub>	V
V <sub>IH</sub>	Input high voltage, except XTAL1, RST, P1.6/SCL, P1.7/SDA	0°C to 70°C		0.2V <sub>DD</sub> +0.9	V <sub>DD</sub> +0.5	V
		-40°C to +85°C		0.2V <sub>DD</sub> +1.0	V <sub>DD</sub> +0.5	V
		-40°C to +125°C		0.2V <sub>DD</sub> +1.0	V <sub>DD</sub> +0.5	V
V <sub>IH1</sub>	Input high voltage, XTAL1, RST	0°C to 70°C		0.7V <sub>DD</sub>	V <sub>DD</sub> +0.5	V
		-40°C to +85°C		0.7V <sub>DD</sub> +0.1	V <sub>DD</sub> +0.5	V
		-40°C to +125°C		0.7V <sub>DD</sub> +0.1	V <sub>DD</sub> +0.5	V
V <sub>IH2</sub>	Input high voltage, P1.6/SCL, P1.7/SDA <sup>3</sup>			0.7V <sub>DD</sub>	6.0	V
V <sub>OL</sub>	Output low voltage, ports 1, 2, 3, except P1.6/SCL, P1.7/SDA <sup>1</sup>		I <sub>OL</sub> = 1.6mA <sup>4</sup>		0.45	V
V <sub>OL1</sub>	Output low voltage, port 0, ALE, PSEN <sup>1</sup>		I <sub>OL</sub> = 3.2mA <sup>4</sup>		0.45	V
V <sub>OL2</sub>	Output low voltage, P1.6/SCL, P1.7/SDA		I <sub>OL</sub> = 3.0mA <sup>4</sup>		0.4	V
V <sub>OH</sub>	Output high voltage, ports 1, 2, 3		V <sub>DD</sub> = 5V ±10%, I <sub>OH</sub> = -60µA I <sub>OH</sub> = -25µA I <sub>OH</sub> = -10µA	2.4		V
				0.75V <sub>DD</sub>		V
				0.9V <sub>DD</sub>		V
V <sub>OH1</sub>	Output high voltage, Port 0 in external bus mode, ALE, PSEN, RST <sup>2</sup>		V <sub>DD</sub> = 5V ±10%, I <sub>OH</sub> = -800µA I <sub>OH</sub> = -300µA I <sub>OH</sub> = -80µA	2.4		V
				0.75V <sub>DD</sub>		V
				0.9V <sub>DD</sub>		V
I <sub>IL</sub>	Logical 0 input current, ports 1, 2, 3, except P1.6/SCL, P1.7/SDA	0°C to 70°C -40°C to +85°C -40°C to +125°C	V <sub>IN</sub> = 0.45V		-50 -75 -75	µA µA µA
I <sub>TL</sub>	Logical 1-to-0 transition current, ports 1, 2, 3, except P1.6/SCL, P1.7/SDA	0°C to 70°C -40°C to +85°C -40°C to +125°C	See note 5		-650 -750 -750	µA µA µA

## CMOS single-chip 8-bit microcontroller 80C528/83C528

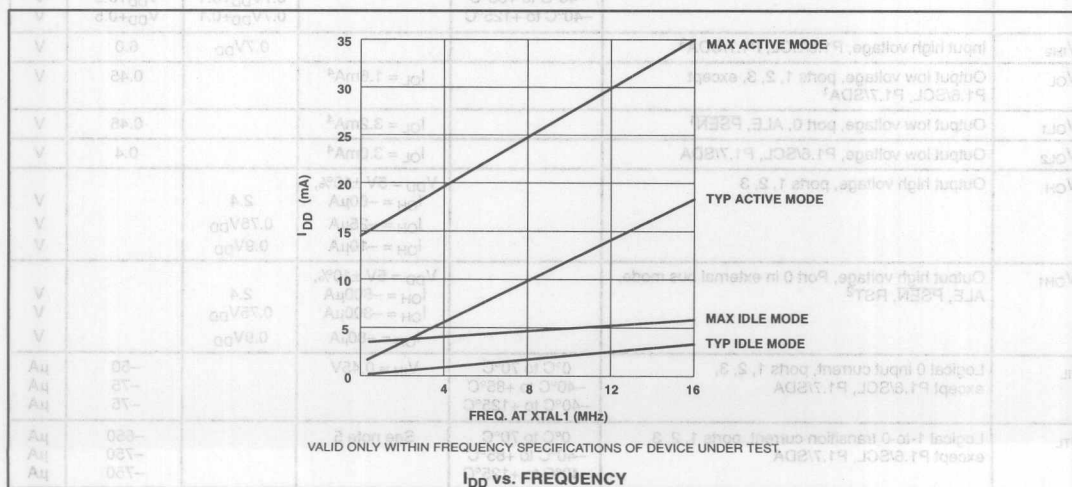
## DC ELECTRICAL CHARACTERISTICS (Continued)

$T_{amb} = 0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$  ( $V_{DD} = 5\text{V} \pm 20\%$ ),  $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$  ( $V_{DD} = 5\text{V} \pm 20\%$ ), or  $-40^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$  ( $V_{DD} = 5\text{V} \pm 10\%$ ),  $V_{SS} = 0\text{V}$

SYMBOL	PARAMETER	PART TYPE	TEST CONDITIONS	LIMITS		UNIT
				MIN	MAX	
$I_{IL1}$	Input leakage current, port 0, EA		$0.45 < V_i < V_{DD}$		$\pm 10$	$\mu\text{A}$
$I_{IL2}$	Input leakage current, P1.6/SCL, P1.7/SDA		$0\text{V} < V_i < 6.0\text{V}$ $0\text{V} < V_{DD} < 6.0\text{V}$		$\pm 10$	$\mu\text{A}$
$I_{DD}$	Power supply current:		See notes 6, 7			
	Active mode				35	mA
	Idle mode				6	mA
	Power down mode				100	$\mu\text{A}$
	Power down mode	$-40^{\circ}\text{C}$ to $+125^{\circ}\text{C}$			150	$\mu\text{A}$
$R_{RST}$	Internal reset pull-down resistor			50	150	k $\Omega$
$C_{IO}$	Capacitance of I/O buffer		Freq. = 1MHz $T_{amb} = 25^{\circ}\text{C}$		10	pF

## NOTES:

- Capacitive loading on Port 0 and Port 2 may cause spurious noise pulses to be superimposed on the LOW level output voltage of ALE, Port 1 and Port 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make a 1-to-0 transition during bus operations. In the worst cases (capacitive loading  $> 100\text{pF}$ ), the noise pulse on the ALE line may exceed 0.8V. In such cases it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.
- Capacitive loading on Port 0 and Port 2 may cause the HIGH level output voltage on ALE and PSEN to momentarily fall below the 0.9 $V_{DD}$  specification when the address bits are stabilizing.
- The input threshold voltage of P1.6 and P1.7 (SIO1) meets the  $I^2\text{C}$  specification, so a voltage below 0.3 $V_{DD}$  will be recognized as a logic 0, while an input above 0.7 $V_{DD}$  will be recognized as a logic 1.
- Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:  
Maximum  $I_{OL}$  per port pin: 10mA  
Maximum  $I_{OL}$  per 8-bit port: —  
Port 0: 26mA  
Ports 1, 2, & 3: 15mA  
Maximum total  $I_{OL}$  for all output pins: 71mA  
If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
- Pins of ports 1, 2, and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when  $V_{IN}$  is approximately 2V.
- See Figures 9 through 12 for  $I_{DD}$  test conditions.
- $I_{DDMAX}$  at other frequencies can be derived from the figure below, where FREQ is the external oscillator frequency in MHz.  
 $I_{DDMAX}$  is given in mA.





## CMOS single-chip 8-bit microcontroller 80C528/83C528

AC ELECTRICAL CHARACTERISTICS<sup>1, 2</sup>

SYMBOL	FIGURE	PARAMETER	16MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
$f_{\text{CLCL}}$	1	Oscillator frequency			1.2	16	MHz
$t_{\text{LHLL}}$	1	ALE pulse width	85		$2t_{\text{CLCL}}-40$		ns
$t_{\text{AVLL}}$	1	Address valid to ALE low	8		$t_{\text{CLCL}}-55$		ns
$t_{\text{LLAX}}$	1	Address hold after ALE low	28		$t_{\text{CLCL}}-35$		ns
$t_{\text{LLIV}}$	1	ALE low to valid instruction in		150		$4t_{\text{CLCL}}-100$	ns
$t_{\text{LLPL}}$	1	ALE low to PSEN low	23		$t_{\text{CLCL}}-40$		ns
$t_{\text{PLPH}}$	1	PSEN pulse width	143		$3t_{\text{CLCL}}-45$		ns
$t_{\text{PLIV}}$	1	PSEN low to valid instruction in		83		$3t_{\text{CLCL}}-105$	ns
$t_{\text{PXIX}}$	1	Input instruction hold after PSEN	0		0		ns
$t_{\text{PXIZ}}$	1	Input instruction float after PSEN		38		$t_{\text{CLCL}}-25$	ns
$t_{\text{AVIV}}$	1	Address to valid instruction in		208		$5t_{\text{CLCL}}-105$	ns
$t_{\text{PLAZ}}$	1	PSEN low to address float		10		10	ns
<b>Data Memory</b>							
$t_{\text{RLRH}}$	2, 3	RD pulse width	275		$6t_{\text{CLCL}}-100$		ns
$t_{\text{WLWH}}$	2, 3	WR pulse width	275		$6t_{\text{CLCL}}-100$		ns
$t_{\text{RLDV}}$	2, 3	RD low to valid data in		148		$5t_{\text{CLCL}}-165$	ns
$t_{\text{RHDZ}}$	2, 3	Data hold after RD	0		0		ns
$t_{\text{RHDZ}}$	2, 3	Data float after RD		55		$2t_{\text{CLCL}}-70$	ns
$t_{\text{LLDV}}$	2, 3	ALE low to valid data in		350		$8t_{\text{CLCL}}-150$	ns
$t_{\text{AVDV}}$	2, 3	Address to valid data in		398		$9t_{\text{CLCL}}-165$	ns
$t_{\text{LLWL}}$	2, 3	ALE low to RD or WR low	138	238	$3t_{\text{CLCL}}-50$	$3t_{\text{CLCL}}+50$	ns
$t_{\text{AVWL}}$	2, 3	Address valid to WR low or RD low	120		$4t_{\text{CLCL}}-130$		ns
$t_{\text{QVWX}}$	2, 3	Data valid to WR transition	3		$t_{\text{CLCL}}-60$		ns
$t_{\text{WHQX}}$	2, 3	Data hold after WR	13		$t_{\text{CLCL}}-50$		ns
$t_{\text{RLAZ}}$	2, 3	RD low to address float		0		0	ns
$t_{\text{WHLH}}$	2, 3	RD or WR high to ALE high	23	103	$t_{\text{CLCL}}-40$	$t_{\text{CLCL}}+40$	ns
<b>External Clock</b>							
$t_{\text{CHCX}}$	6	High time	20		20		ns
$t_{\text{CLCX}}$	6	Low time	20		20		ns
$t_{\text{CLCH}}$	6	Rise time		20		20	ns
$t_{\text{CHCL}}$	6	Fall time		20		20	ns
<b>Shift Register</b>							
$t_{\text{XLXL}}$	4	Serial port clock cycle time	750		$12t_{\text{CLCL}}$		ns
$t_{\text{QVXH}}$	4	Output data setup to clock rising edge	492		$10t_{\text{CLCL}}-133$		ns
$t_{\text{XHGX}}$	4	Output data hold after clock rising edge	8		$2t_{\text{CLCL}}-117$		ns
$t_{\text{XHDX}}$	4	Input data hold after clock rising edge	0		0		ns
$t_{\text{XHDX}}$	4	Clock rising edge to input data valid		492		$10t_{\text{CLCL}}-133$	ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.



## CMOS single-chip 8-bit microcontroller 80C528/83C528

## AC ELECTRICAL CHARACTERISTICS – I<sup>2</sup>C INTERFACE

SYMBOL	PARAMETER	INPUT	OUTPUT	I <sup>2</sup> C SPECIFICATION
<b>SCL TIMING CHARACTERISTICS</b>				
t <sub>HD</sub> : STA	START condition hold time	≥ 14 t <sub>CLCL</sub> <sup>1</sup>	Note 2	≥ 4.0μs
t <sub>LOW</sub>	SCL LOW time	≥ 16 t <sub>CLCL</sub>	Note 2	≥ 4.7μs
t <sub>HIGH</sub>	SCL HIGH time	≥ 14 t <sub>CLCL</sub> <sup>1</sup>	≥ 80 t <sub>CLCL</sub> <sup>3</sup>	≥ 4.0μs
t <sub>RC</sub>	SCL rise time	≤ 1μs <sup>4</sup>	Note 5	≤ 1.0μs
t <sub>FC</sub>	SCL fall time	≤ 0.3μs <sup>4</sup>	≤ 0.3μs <sup>6</sup>	≤ 0.3μs
<b>SDA TIMING CHARACTERISTICS</b>				
t <sub>SU</sub> : DAT1	Data set-up time	≥ 250ns	Note 2	≥ 250ns
t <sub>HD</sub> : DAT	Data hold time	≥ 0ns	Note 2	≥ 0ns
t <sub>SU</sub> : STA	Repeated START set-up time	≥ 14 t <sub>CLCL</sub> <sup>1</sup>	Note 2	≥ 4.7μs
t <sub>SU</sub> : STO	STOP condition set-up time	≥ 14 t <sub>CLCL</sub> <sup>1</sup>	Note 2	≥ 4.0μs
t <sub>BUF</sub>	Bus free time	≥ 14 t <sub>CLCL</sub> <sup>1</sup>	Note 2	≥ 4.7μs
t <sub>RD</sub>	SDA rise time	≤ 1μs <sup>4</sup>	Note 5	≤ 1.0μs
t <sub>FD</sub>	SDA fall time	≤ 0.3μs <sup>4</sup>	≤ 0.3μs <sup>6</sup>	≤ 0.3μs

NOTES:

- At  $f_{CLK} = 3.5\text{MHz}$ , this evaluates to  $14 \times 286\text{ns} = 4\mu\text{s}$ , i.e., the bit-level I<sup>2</sup>C interface can respond to the I<sup>2</sup>C protocol for  $f_{CLK} \geq 3.5\text{MHz}$ .
- This parameter is determined by the user software, it has to comply with the I<sup>2</sup>C.
- This value gives the autotclock pulse length which meets the I<sup>2</sup>C specification for the specified XTAL clock frequency range. Alternatively, the SCL pulse may be timed by software.
- Spikes on SDA and SCL lines with a duration of less than  $4 \times f_{CLK}$  will be filtered out.
- The rise time is determined by the external bus line capacitance and pull-up resistor, it must be  $\leq 1\mu\text{s}$ .
- The maximum capacitance on bus lines SDA and SCL is 400pF.

## 80C528/83C528

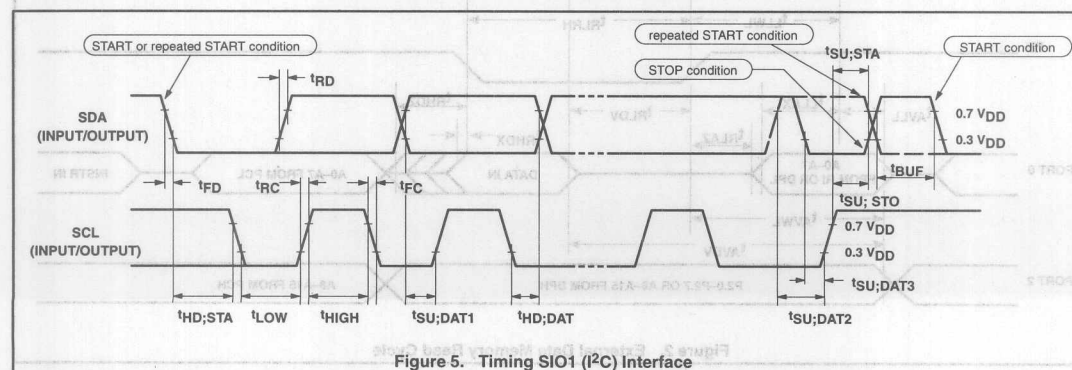
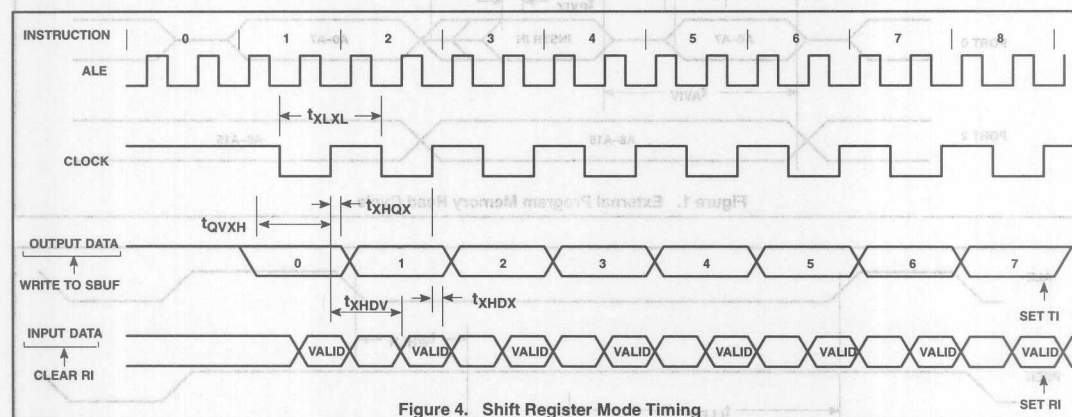
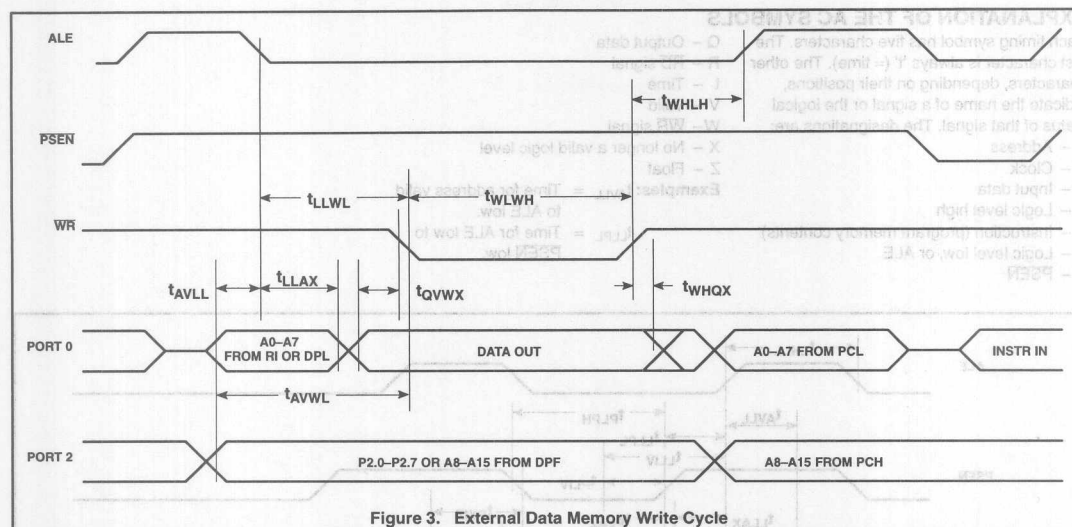
Each timing symbol has five characters. The Q

- A – Address
- C – Clock
- D – Input data
- H – Logic level high
- I – Instruction (program memory contents)
- L – Logic level low, or ALE
- P – PSEN

**Examples:**  $t_{AVLL}$  = Time for address valid to ALE low.  
 $t_{LLPL}$  = Time for ALE low to PSEN low.



## CMOS single-chip 8-bit microcontroller 80C528/83C528



## CMOS single-chip 8-bit microcontroller 80C528/83C528

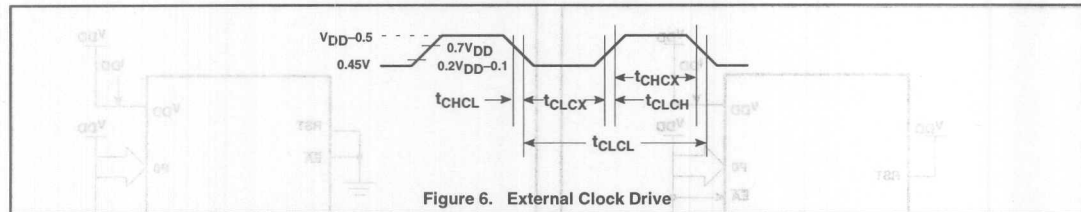


Figure 6. External Clock Drive

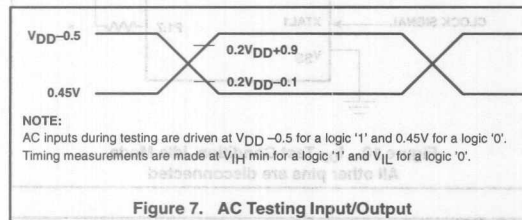


Figure 7. AC Testing Input/Output

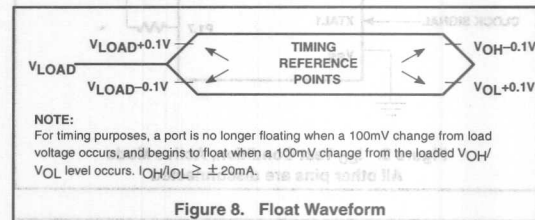
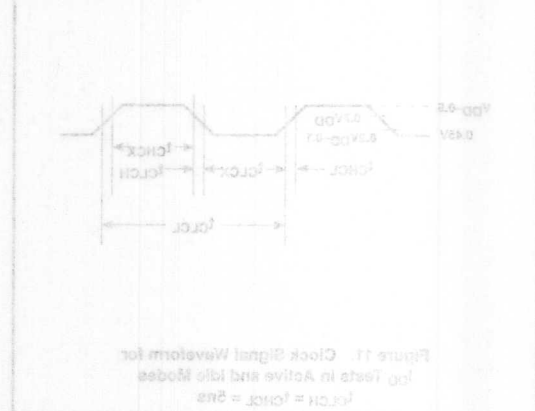
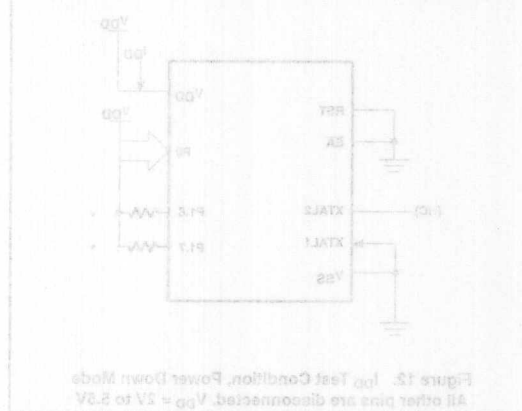


Figure 8. Float Waveform

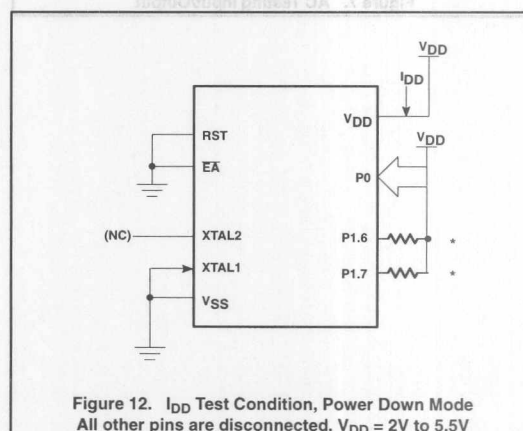
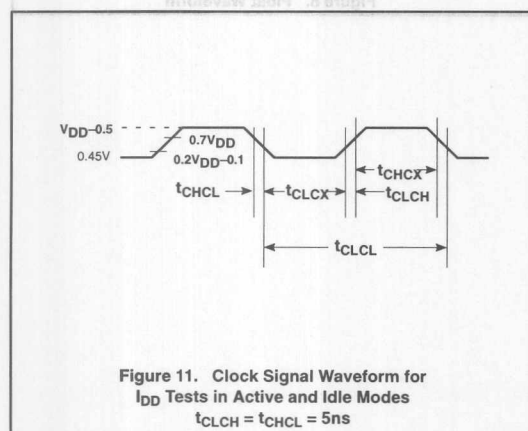
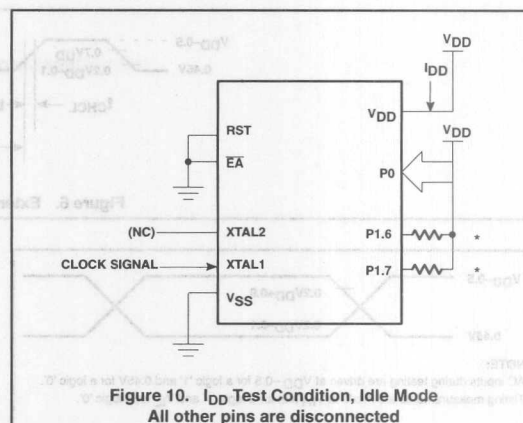
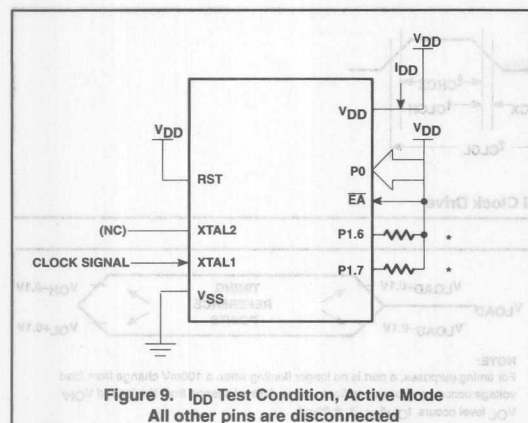


NOTE:  
Ports 0, 1, and 2 should be connected to  $V_{DD}$  through resistors of sufficiently high value such that the sink current into these pins does not exceed the I/O specifications.  
tCHCL = tCHCX = 5ns

Purchase of Philips' iC components conveys a license under the Philips' iC patent to use the components in the iC-system provided the system conforms to the iC specifications defined by Philips.



## CMOS single-chip 8-bit microcontroller 80C528/83C528

**NOTE:**

- \* Ports 1.6 and 1.6 should be connected to  $V_{DD}$  through resistors of sufficiently high value such that the sink current into these pins does not exceed the  $I_{OL1}$  specifications.



Purchase of Philips' I<sup>2</sup>C components conveys a license under the Philips' I<sup>2</sup>C patent to use the components in the I<sup>2</sup>C-system provided the system conforms to the I<sup>2</sup>C specifications defined by Philips.



# CMOS single-chip 8-bit microcontroller 87C528

## DESCRIPTION

The 87C528 single-chip 8-bit microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 87C528 has the same instruction set as the 80C51. Three versions of the derivative exist:

- 83C528—32k bytes mask programmable ROM
- 80C528—ROMless version of the 83C528
- 87C528—32k bytes EPROM

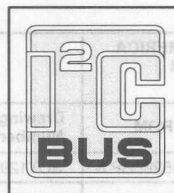
This device provides architectural enhancements that make it applicable in a variety of applications in consumer, telecom and general control systems, especially in those systems which need large ROM and RAM capacity on-chip.

The 87C528 contains a 32k × 8 EPROM, a 512 × 8 RAM, four 8-bit I/O ports, two 16-bit timer/event counters (identical to the timers of the 80C51), a 16-bit timer (identical to the timer 2 of the 80C52), a watchdog timer with a separate oscillator, a multi-source, two-priority-level, nested interrupt structure, two serial interfaces (UART and I<sup>2</sup>C-bus), and on-chip oscillator and timing circuits.

In addition, the 87C528 has two software selectable modes of power reduction—idle mode and power-down mode. The idle mode freezes the CPU while allowing the RAM, timers, serial port, and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.

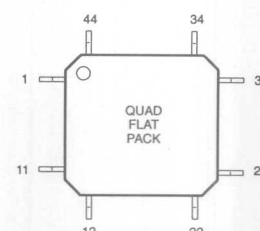
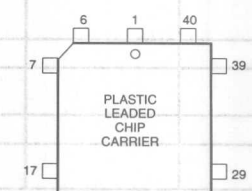
## FEATURES

- 80C51 instruction set
  - 32k × 8 EPROM
  - 512 × 8 RAM
  - Memory addressing capability 64k ROM and 64k RAM
  - Three 16-bit counter/timers
  - On-chip watchdog timer with oscillator
  - Full duplex UART
  - I<sup>2</sup>C serial interface
- Power control modes:
  - Idle mode
  - Power-down mode
  - Warm start from power-down
- CMOS and TTL compatible
- Extended temperature ranges
- EPROM code protection
- OTP package available
- Two speed ranges at V<sub>CC</sub> = 5V
  - 16MHz
  - 20MHz



## PIN CONFIGURATIONS

T2/P1.0	1	40	VDD
T2EX/P1.1	2	39	P0.0/AD0
P1.2	3	38	P0.1/AD1
P1.3	4	37	P0.2/AD2
P1.4	5	36	P0.3/AD3
P1.5	6	35	P0.4/AD4
SCL/P1.6	7	34	P0.5/AD5
SDA/P1.7	8	33	P0.6/AD6
RST	9	32	P0.7/AD7
RxD/P3.0	10	31	EA
TxD/P3.1	11	30	ALE
INT0/P3.2	12	29	PSEN
INT1/P3.3	13	28	P2.7/A15
T0/P3.4	14	27	P2.6/A14
T1/P3.5	15	26	P2.5/A13
WR/P3.6	16	25	P2.4/A12
RD/P3.7	17	24	P2.3/A11
XTAL2	18	23	P2.2/A10
XTAL1	19	22	P2.1/A9
VSS	20	21	P2.0/A8



## CMOS single-chip 8-bit microcontroller 87C528

## ORDERING INFORMATION

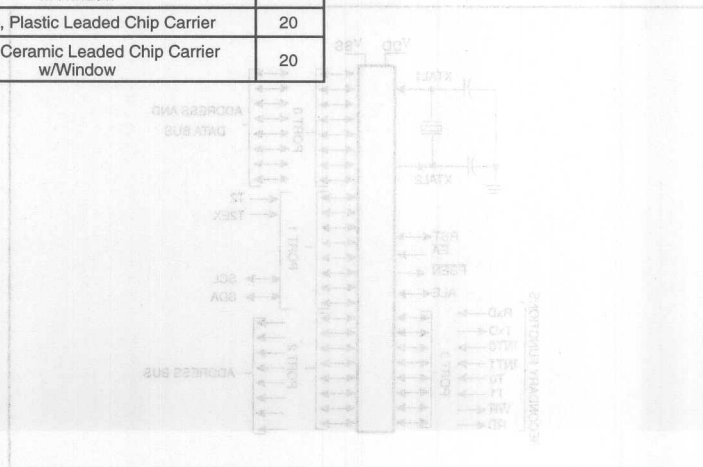
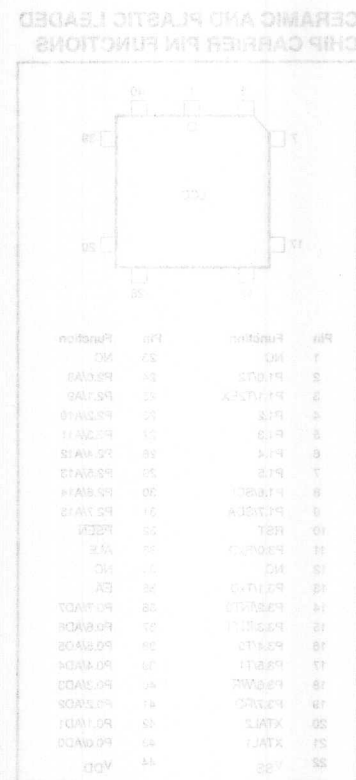
[illegible]

**NOTE:**

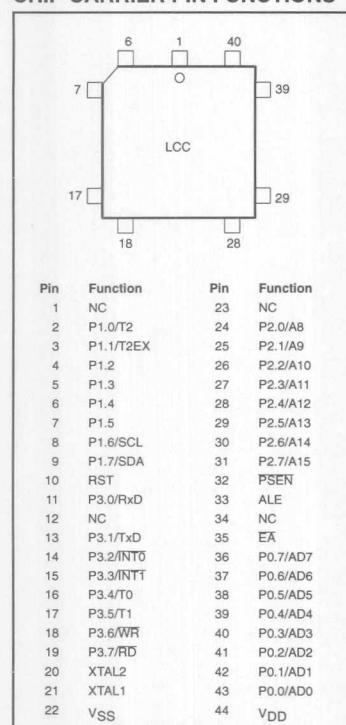
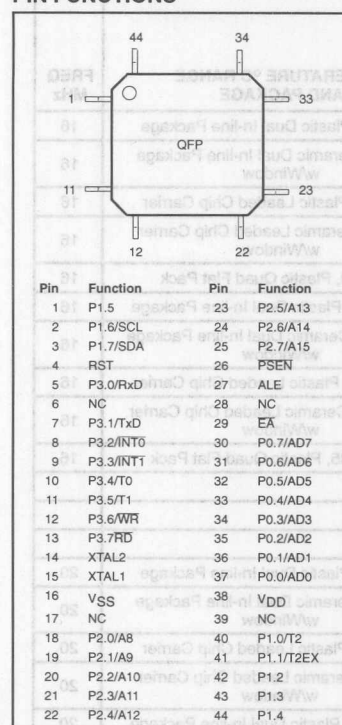
1. xxx denotes the ROM code number.
2. SOT311 replaced by SOT307-2.

## CMOS single-chip 8-bit microcontroller 87C528

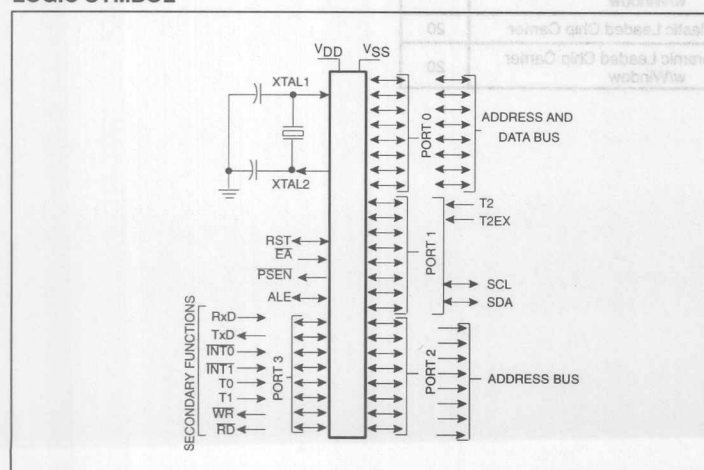
EPROM	Drawing Number	TEMPERATURE °C RANGE AND PACKAGE	FREQ MHz
P87C528EBP N	0415C	0 to +70, Plastic Dual In-line Package	16
P87C528EBF FA	0590B	0 to +70, Ceramic Dual In-line Package w/Window	16
P87C528EBA A	0403G	0 to +70, Plastic Leaded Chip Carrier	16
P87C528EBL KA	1472A	0 to +70, Ceramic Leaded Chip Carrier w/Window	16
P87C528EBB B	SOT307-2 <sup>2</sup>	0 to +70, Plastic Quad Flat Pack	16
P87C528EFP N	0415C	-40 to +85, Plastic Dual In-line Package	16
P87C528EFF FA	0590B	-40 to +85, Ceramic Dual In-line Package w/Window	16
P87C528EFA A	0403G	-40 to +85, Plastic Leaded Chip Carrier	16
P87C528EFL KA	1472A	-40 to +85, Ceramic Leaded Chip Carrier w/Window	16
P87C528EFB B	SOT307-2 <sup>2</sup>	-40 to +85, Plastic Quad Flat Pack	16
P87C528GBP N	0415C	0 to +70, Plastic Dual In-line Package	20
P87C528GBF FA	0590B	0 to +70, Ceramic Dual In-line Package w/Window	20
P87C528GBA A	0403G	0 to +70, Plastic Leaded Chip Carrier	20
P87C528GBL KA	1472A	0 to +70, Ceramic Leaded Chip Carrier w/Window	20
P87C528GFP N	0415C	-40 to +85, Plastic Dual In-line Package	20
P87C528GFF FA	0590B	0 to +70, Ceramic Dual In-line Package w/Window	20
P87C528GFA A	0403G	0 to +70, Plastic Leaded Chip Carrier	20
P87C528GFL KA	1472A	0 to +70, Ceramic Leaded Chip Carrier w/Window	20



## CMOS single-chip 8-bit microcontroller 87C528

CERAMIC AND PLASTIC LEADED  
CHIP CARRIER PIN FUNCTIONSPLASTIC QUAD FLAT PACK  
PIN FUNCTIONS

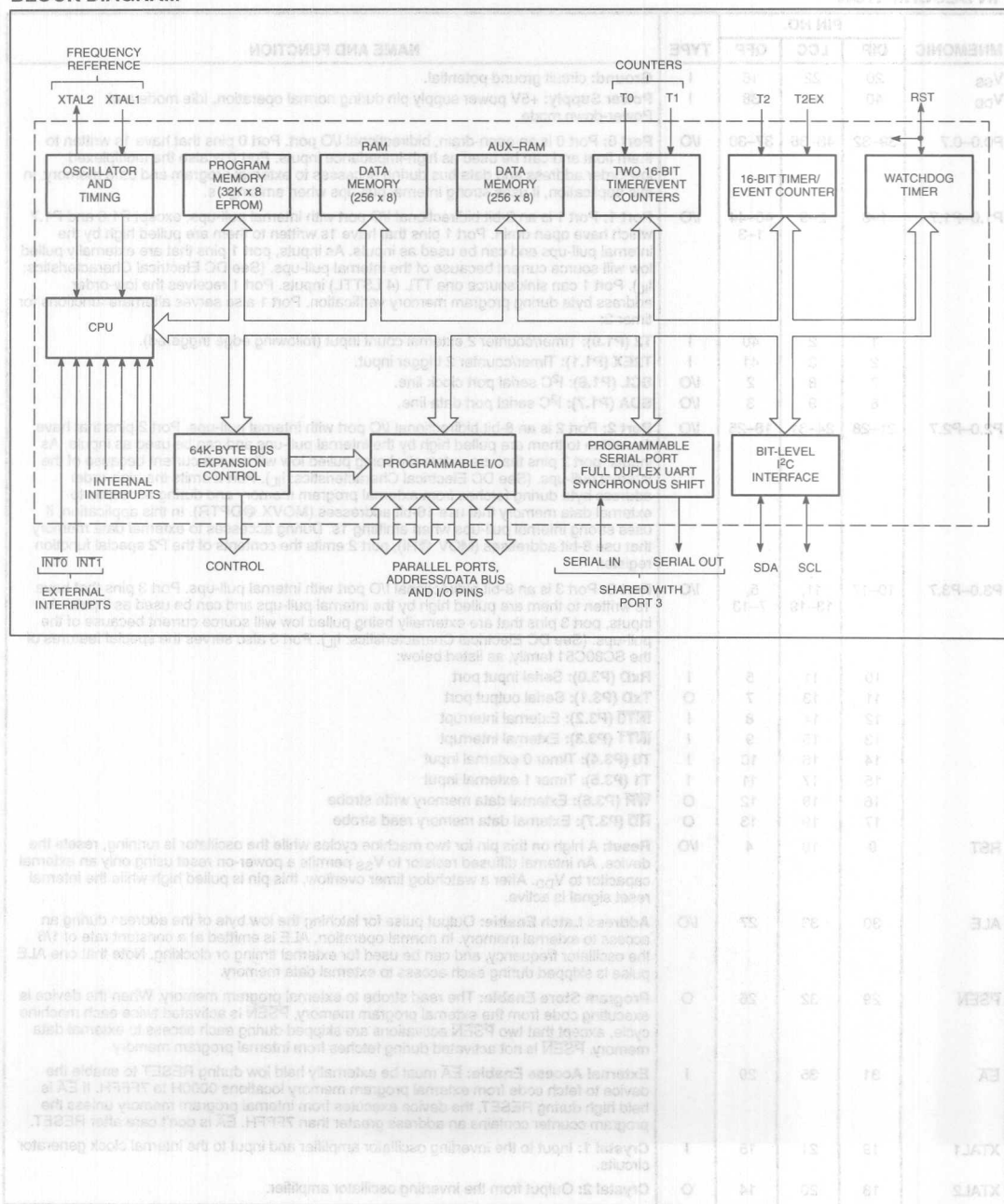
## LOGIC SYMBOL



## CMOS single-chip 8-bit microcontroller

87C528

### BLOCK DIAGRAM





## CMOS single-chip 8-bit microcontroller 87C528

## PIN DESCRIPTION

MNEMONIC	PIN NO.			TYPE	NAME AND FUNCTION
	DIP	LCC	QFP		
V <sub>SS</sub>	20	22	16	I	<b>Ground:</b> circuit ground potential.
V <sub>DD</sub>	40	44	38	I	<b>Power Supply:</b> +5V power supply pin during normal operation, Idle mode and Power-down mode.
P0.0–P0.7	39–32	43–36	37–30	I/O	<b>Port 0:</b> Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s.
P1.0–P1.7	1–8	2–9	40–44 1–3	I/O	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pull-ups, except P1.6 and P1.7 which have open drain. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 1 can sink/source one TTL (4 LSTTL) inputs. Port 1 receives the low-order address byte during program memory verification. Port 1 also serves alternate functions for timer 2:  <b>T2 (P1.0):</b> Timer/counter 2 external count input (following edge triggered). <b>T2EX (P1.1):</b> Timer/counter 2 trigger input. <b>SCL (P1.6):</b> I <sup>2</sup> C serial port clock line. <b>SDA (P1.7):</b> I <sup>2</sup> C serial port data line.
P2.0–P2.7	21–28	24–31	18–25	I/O	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register.
P3.0–P3.7	10–17	11, 13–19	5, 7–13	I/O	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 3 also serves the special features of the SC80C51 family, as listed below:  <b>RxD (P3.0):</b> Serial input port <b>TxD (P3.1):</b> Serial output port <b>INT0 (P3.2):</b> External interrupt <b>INT1 (P3.3):</b> External interrupt <b>T0 (P3.4):</b> Timer 0 external input <b>T1 (P3.5):</b> Timer 1 external input <b>WR (P3.6):</b> External data memory write strobe <b>RD (P3.7):</b> External data memory read strobe
RST	9	10	4	I/O	<b>Reset:</b> A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to V <sub>SS</sub> permits a power-on reset using only an external capacitor to V <sub>DD</sub> . After a watchdog timer overflow, this pin is pulled high while the internal reset signal is active.
ALE	30	33	27	I/O	<b>Address Latch Enable:</b> Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory.
PSEN	29	32	26	O	<b>Program Store Enable:</b> The read strobe to external program memory. When the device is executing code from the external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory. PSEN is not activated during fetches from internal program memory.
EA	31	35	29	I	<b>External Access Enable:</b> EA must be externally held low during RESET to enable the device to fetch code from external program memory locations 0000H to 7FFFH. If EA is held high during RESET, the device executes from internal program memory unless the program counter contains an address greater than 7FFFH. EA is don't care after RESET.
XTAL1	19	21	15	I	<b>Crystal 1:</b> Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	20	14	O	<b>Crystal 2:</b> Output from the inverting oscillator amplifier.

## CMOS single-chip 8-bit microcontroller 87C528

Table 1. 8XC524/8XC528 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			LSB								
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
DPTR: DPH DPL	Data pointer (2 bytes): Data pointer high Data pointer low	83H 82H									00H 00H
			AF	AE	AD	AC	AB	AA	A9	A8	
IE*#	Interrupt enable	A8H	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0	00H
IP*#	Interrupt priority	B8H	—	PS1	PT2	PS0	PT1	PX1	PT0	PX0	x0000000B
P0*	Port 0	80H	87	86	85	84	83	82	81	80	
			AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
P1*	Port 1	90H	97	96	95	94	93	92	91	90	
			SDA	SEL	—	—	—	—	T2EX	T2	FFH
P2*	Port 2	A0H	A7	A6	A5	A4	A3	A2	A1	A0	
			A15	A14	A13	A12	A11	A10	A9	A8	FFH
P3*	Port 3	B0H	B7	B6	B5	B4	B3	B2	B1	B0	
			RD	WR	T1	T0	INT1	INT0	TxD	RxD	FFH
PCON	Power control	87H	SMOD	—	—	—	GF1	GF0	PD	IDL	0xxx0000B
PSW*	Program status word	D0H	D7	D6	D5	D4	D3	D2	D1	D0	
			CY	AC	F0	RS1	RS0	OV	F1	P	00H
RCAP2H# RCAP2L# SBUF	Capture high Capture low Serial data buffer	CBH CAH 99H	9F	9E	9D	9C	9B	9A	99	98	00H 00H xxxxxxx0B
			SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00H
SCON*	Serial controller	98H	SDI	0	0	0	0	0	0	0	x0000000B
			SD0	X	X	X	X	X	X	X	0xxxxxxxB
S1BIT#	Serial I <sup>2</sup> C data	D9H/RD	INT	X	X	X	X	X	X	X	0xxxxxxxB
			DF	DE	DD	DC	DB	DA	D9	D8	
S1SCS*#	Serial I <sup>2</sup> C control	D8H/RD	SDI	SCI	CLH	BB	RBF	WBF	STR	ENS	xxxx0000B
			SD0	SC0	CLH	X	X	X	STR	ENS	00xxxx00B
SP	Stack pointer	81H	8F	8E	8D	8C	8B	8A	89	88	07H
			TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
TCON*	Timer control	88H	CF	CE	CD	CC	CB	CA	C9	C8	
			TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	00H
T2CON*#	Timer 2 control	C8H									00H
TH0	Timer high 0	8CH									00H
TH1	Timer high 1	8DH									00H
TH2#	Timer high 2	CDH									00H
TL0	Timer low 0	8AH									00H
TL1	Timer low 1	8BH									00H
TL2#	Timer low 2	CCH									00H
T3#	Watchdog timer	FFH									00H
TMOD	Timer mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H
WDCON#	Watchdog control	A5H									A5H

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.

## CMOS single-chip 8-bit microcontroller 87C528

Table 2. Internal and External Program Memory Access with Security Bit Set

INSTRUCTION	ACCESS TO INTERNAL PROGRAM MEMORY	ACCESS TO EXTERNAL PROGRAM MEMORY
MOVC in internal program memory	YES	YES
MOVC in external program memory	NO	YES

**INTERNAL DATA MEMORY**

The internal data memory is divided into three physically separated segments: 256 bytes of RAM, 256 bytes of AUX-RAM, and a 128 bytes special function area. These can be addressed each in a different way.

- RAM 0 to 127 can be addressed directly and indirectly as in the 80C51. Address pointers are R0 and R1 of the selected register bank.
- RAM 128 to 255 can only be addressed indirectly as in the 80C51. Address pointers are R0 and R1 of the selected register bank.
- AUX-RAM 0 to 255 is indirectly addressed in the same way as external data memory with the MOVX instructions. Address pointers are R0, R1 of the selected register bank and DPTR. An access to AUX-RAM 0 to 255 will not affect ports P0, P2, P3.6 and P3.7.

An access to external data memory locations higher than 255 will be performed with the MOVX DPTR instructions in the same way as in the 8051 structure, so with P0 and P2 as data/address bus and P3.6 and P3.7 as write and read timing signals. Note that these external data memory cannot be accessed with R0 and R1 as address pointer.

**TIMER 2**

Timer 2 is functionally equal to the Timer 2 of the 8052AH. Timer 2 is a 16-bit timer/counter. These 16 bits are formed by two special function registers TL2 and TH2. Another pair of special function register RCAP2L and RCAP2H form a 16-bit capture register or a 16-bit reload register. Like Timer 0 and 1, it can operate either as a timer or as an event counter. This is selected by bit C/T2N in the special function register T2CON. It has three operating modes: capture, autoloader, and baud rate generator mode which are selected by bits in T2CON.

**WATCHDOG TIMER T3**

The watchdog timer consists of an 11-bit prescaler and an 8-bit timer formed by special function register T3. The prescaler is incremented by an on-chip oscillator with a fixed frequency of 1MHz. The maximum tolerance on this frequency is -50% and +100%. The 8-bit timer increments every 2048 cycles of the on-chip oscillator. When a timer overflow occurs, the microcontroller is

reset and a reset output pulse of  $16 \times 2048$  cycles of the on-chip oscillator is generated at pin RST. The internal RESET signal is not inhibited when the external RST pin is kept low by, for example, an external reset circuit. The RESET signal drives port 1, 2, 3 into the high state and port 0 into the high impedance state.

The watchdog timer is controlled by one special function register WDCON with the direct address location A5H. WDCON can be read and written by software. A value of A5H in WDCON halts the on-chip oscillator and clears both the prescaler and timer T3. After the RESET signal, WDCON contains A5H. Every value other than A5H in WDCON enables the watchdog timer. When the watchdog timer is enabled, it runs independently of the XTAL-clock.

Timer T3 can be read on the fly. Timer T3 can only be written if WDCON contains the value 5AH. A successful write operation to T3 will clear the prescaler and WDCON, leaving the watchdog enabled and preventing inadvertent changes of T3. To prevent an overflow of the watchdog timer, the user program has to reload the watchdog timer within periods that are shorter than the programmed watchdog timer interval. This time interval is determined by an 8-bit value that has to be loaded in register T3 while at the same time the prescaler is cleared by hardware.

Watchdog timer interval =  

$$\frac{[256 - (T3)] \times 2048}{\text{on-chip oscillator frequency}}$$

**BIT-LEVEL I<sup>2</sup>C INTERFACE**

This bit-level serial I/O interface supports the I<sup>2</sup>C-bus. P1.6/SCL and P1.7/SDA are the serial I/O pins. These two pins meet the I<sup>2</sup>C specification concerning the input levels and output drive capability. Consequently, these pins have an open drain output configuration. All the four modes of the I<sup>2</sup>C-bus are supported:

- master transmitter
- master receiver
- slave transmitter
- slave receiver

The advantages of the bit-level I<sup>2</sup>C hardware compared with a full software I<sup>2</sup>C implementation are:

- the hardware can generate the SCL pulse

- Testing a single bit (RBF respectively, WBF) is sufficient as a check for error free transmission.

The bit-level I<sup>2</sup>C hardware operates on serial bit level and performs the following functions:

- filtering the incoming serial data and clock signals
- recognizing the START condition
- generating a serial interrupt request SI after reception of a START condition and the first falling edge of the serial clock
- recognizing the STOP condition
- recognizing a serial clock pulse on the SCL line
- latching a serial bit on the SDA line (SDI)
- stretching the SCL LOW period of the serial clock to suspend the transfer of the next serial data bit
- setting Read Bit Finished (RBF) when the SCL clock pulse has finished and Write Bit Finished (WBF) if there is no arbitration loss detected (i.e., SDA = 0 while SDO = 1)
- setting a serial clock Low-to-High detected (CLH) flag
- setting a Bus Busy (BB) flag on a START condition and clearing this flag on a STOP condition
- releasing the SCL line and clearing the CLH, RBF and WBF flags to resume transfer of the next serial data bit
- generating an automatic clock if the single bit data register S1BIT is used in master mode.

The following functions must be done in software:

- handling the I<sup>2</sup>C START interrupts
- converting serial to parallel data when receiving
- converting parallel to serial data when transmitting
- comparing the received slave address with its own
- interpreting the acknowledge information
- guarding the I<sup>2</sup>C status if RBF or WBF = 0.

Additionally, if acting as master:

- generating START and STOP conditions
- handling bus arbitration
- generating serial clock pulses if S1BIT is not used.

Three SFRs control the bit-level I<sup>2</sup>C interface: S1INT, S1BIT and S1SCS.

## CMOS single-chip 8-bit microcontroller

87C528

## OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator, as shown in the Logic Symbol.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

## RESET

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. To insure a good power-up reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few milliseconds) plus two machine cycles. At power-up, the voltage on  $V_{DD}$  and RST must come up at the same time for a proper start-up.

## IDLE MODE

In idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay

active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

## POWER-DOWN MODE

In the power-down mode, the oscillator is stopped and the instruction to invoke power-down is the last instruction executed. The power-down mode can be terminated by a RESET in the same way as in the 80C51 or in addition by one of two external interrupts, INT0 or INT1. A termination with an external interrupt does not affect the internal data memory and does not affect the special function registers. This makes it possible to exit power-down without changing the port output levels. To terminate the power-down mode with an external interrupt INT0 or INT1 must be switched to level-sensitive and must be enabled. The external interrupt input signal INT0 and INT1 must be kept low until

the oscillator has restarted and stabilized. An instruction following the instruction that puts the device in the power-down mode will be executed. A reset generated by the watchdog timer terminates the power-down mode in the same way as an external RESET, and only the contents of the on-chip RAM are preserved. The control bits for the reduced power modes are in the special function register PCON.

## DESIGN CONSIDERATIONS

At power-on, the voltage on  $V_{DD}$  and RST must come up at the same time for a proper start-up.

When the idle mode is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write when idle is terminated by reset, the instruction following the one that invokes idle should not be one that writes to a port pin or to external memory.

Table 3 shows the state of I/O ports during low current operating modes.

**Table 3. External Pin Status During Idle and Power-Down Modes**

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

## ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>

PARAMETER	RATING	UNIT
Operating temperature under bias	0 to +70, or -40 to +85, or -40 to +125	°C
Storage temperature range	-65 to +150	°C
Voltage on any other pin to $V_{SS}$	-0.5 to $V_{DD} + 0.5$	V
Input, output current on any two pins	±10	mA
Power dissipation (based on package heat transfer limitations, not device power consumption)	1.0	W

### NOTES:

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
- This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
- Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to  $V_{SS}$  unless otherwise noted.



## CMOS single-chip 8-bit microcontroller

87C528

## DC ELECTRICAL CHARACTERISTICS

 $T_{amb} = 0^{\circ}\text{C to } +70^{\circ}\text{C}$  ( $V_{DD} = 5\text{V} \pm 20\%$ ),  $-40^{\circ}\text{C to } +85^{\circ}\text{C}$  ( $V_{DD} = 5\text{V} \pm 20\%$ ), or  $-40^{\circ}\text{C to } +125^{\circ}\text{C}$  ( $V_{DD} = 5\text{V} \pm 10\%$ ),  $V_{SS} = 0\text{V}$ 

SYMBOL	PARAMETER	PART TYPE	TEST CONDITIONS	LIMITS		UNIT
				MIN	MAX	
$V_{IL}$	Input low voltage, except EA, P1.6/SCL, P1.7/SDA	$0^{\circ}\text{C to } 70^{\circ}\text{C}$ $-40^{\circ}\text{C to } +85^{\circ}\text{C}$		-0.5 -0.5	$0.2V_{CC}-0.1$ $0.2V_{CC}-0.15$	V
$V_{IL1}$	Input low voltage to EA	$0^{\circ}\text{C to } 70^{\circ}\text{C}$ $-40^{\circ}\text{C to } +85^{\circ}\text{C}$		0 0	$0.2V_{CC}-0.3$ $0.2V_{CC}-0.35$	V
$V_{IL2}$	Input low voltage to P1.6/SCL, P1.7/SDA <sup>5</sup>			-0.5	0.3V	V
$V_{IH}$	Input high voltage, except XTAL1, RST, P1.6/SCL, P1.7/SDA	$0^{\circ}\text{C to } 70^{\circ}\text{C}$ $-40^{\circ}\text{C to } +85^{\circ}\text{C}$		$0.2V_{CC}+0.9$ $0.2V_{CC}+1.0$	$V_{CC}+0.5$ $V_{CC}+0.5$	V
$V_{IH1}$	Input high voltage, XTAL1, RST	$0^{\circ}\text{C to } 70^{\circ}\text{C}$ $-40^{\circ}\text{C to } +85^{\circ}\text{C}$		$0.7V_{CC}$ $0.7V_{CC}+0.1$	$V_{CC}+0.5$ $V_{CC}+0.5$	V
$V_{IH2}$	Input high voltage, P1.6/SCL, P1.7/SDA <sup>5</sup>			3.0	6.0	V
$V_{OL}$	Output low voltage, ports 1, 2, 3, except P1.6/SCL, P1.7/SDA <sup>1</sup>		$I_{OL} = 1.6\text{mA}^1$		0.45	V
$V_{OL1}$	Output low voltage, port 0, ALE, PSEN <sup>1</sup>		$I_{OL} = 3.2\text{mA}^1$		0.45	V
$V_{OL2}$	Output low voltage, P1.6/SCL, P1.7/SDA		$I_{OL} = 3.0\text{mA}^1$		0.4	V
$V_{OH}$	Output high voltage, ports 1, 2, 3		$I_{OH} = -60\mu\text{A}$ $I_{OH} = -25\mu\text{A}$	2.4 $0.75V_{CC}$		V
$V_{OH1}$	Output high voltage, Port 0 in external bus mode, ALE, PSEN, RST		$I_{OH} = -800\mu\text{A}$ $I_{OH} = -300\mu\text{A}$	2.4 $0.75V_{CC}$		V
$I_{IL}$	Logical 0 input current, ports 1, 2, 3, except P1.6/SCL, P1.7/SDA	$0^{\circ}\text{C to } 70^{\circ}\text{C}$ $-40^{\circ}\text{C to } +85^{\circ}\text{C}$	$V_{IN} = 0.45\text{V}$		-50 -75	$\mu\text{A}$ $\mu\text{A}$
$I_{TL}$	Logical 1-to-0 transition current, ports 1, 2, 3, except P1.6/SCL, P1.7/SDA	$0^{\circ}\text{C to } 70^{\circ}\text{C}$ $-40^{\circ}\text{C to } +85^{\circ}\text{C}$	See note 3		-650 -750	$\mu\text{A}$ $\mu\text{A}$
$I_{IL1}$	Input leakage current, port 0		$V_{IN} = V_{IL}$ or $V_{IH}$		$\pm 10$	$\mu\text{A}$
$I_{IL2}$	Input leakage current, P1.6/SCL, P1.7/SDA		$0\text{V} < V_i < 6.0\text{V}$ $0\text{V} < V_{CC} < 6.0\text{V}$		$\pm 10$	$\mu\text{A}$ $\mu\text{A}$
$I_{CC}$	Power supply current:		See note 4			
	Active mode @ 16MHz	$0^{\circ}\text{C to } 70^{\circ}\text{C}$ $-40^{\circ}\text{C to } +85^{\circ}\text{C}$			25 35	mA
	Idle mode @ 16MHz	$0^{\circ}\text{C to } 70^{\circ}\text{C}$ $-40^{\circ}\text{C to } +85^{\circ}\text{C}$			5 6	mA
	Power down mode				50	$\mu\text{A}$
$R_{RST}$	Internal reset pull-down resistor			50	300	k $\Omega$
$C_{IO}$	Pin Capacitance				10	pF

## NOTES:

- Capacitive loading on Port 0 and Port 2 may cause spurious noise pulses to be superimposed on the  $V_{OL}$ s of ALE and ports 1 and 3. The noise is due to external bus capacitance discharging into the port and port 2 pins when these pins make 1-to-0 transactions during bus operations. In the worst cases (capacitive loading  $> 100\text{pF}$ ), the noise pulse on the ALE pin may exceed 0.8V. In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input. Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows: 10mA per port pin, port 0 total (all bits) 26mA, ports 1, 2, and total each (all bits) 15mA.
- Capacitive loading on Ports 0 and 2 may cause the  $V_{OH}$  on ALE and PSEN to momentarily fall below the  $0.9V_{CC}$  specification when the address bits are stabilizing.
- Pins of ports 1, 2, and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when  $V_{IN}$  is approximately 2V.
- See Figures 10 through 13 for  $I_{CC}$  test conditions.
- The input threshold voltage of P1.6 and P1.7 (SIO1) meets the  $I^2\text{C}$  specification, so an input voltage below 1.5V will be recognized as a logic 0 while an input voltage above 3.0V will be recognized as a logic 1.



## CMOS single-chip 8-bit microcontroller 87C528

AC ELECTRICAL CHARACTERISTICS<sup>1, 2</sup>

SYMBOL	FIGURE	PARAMETER	16MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
$1/t_{CLCL}$	1	Oscillator frequency: 87C528 Speed Versions P878C528EXX P878C528GXX			3.5 3.5	16 20	MHz MHz
$t_{LHLL}$	1	ALE pulse width	85		$2t_{CLCL}-40$		ns
$t_{AVLL}$	1	Address valid to ALE low	8		$t_{CLCL}-55$		ns
$t_{LLAX}$	1	Address hold after ALE low	28		$t_{CLCL}-35$		ns
$t_{LLIV}$	1	ALE low to valid instruction in		150		$4t_{CLCL}-100$	ns
$t_{LLPL}$	1	ALE low to PSEN low	23		$t_{CLCL}-40$		ns
$t_{PLPH}$	1	PSEN pulse width	143		$3t_{CLCL}-45$		ns
$t_{PLIV}$	1	PSEN low to valid instruction in		83		$3t_{CLCL}-105$	ns
$t_{PXIX}$	1	Input instruction hold after PSEN	0		0		ns
$t_{PXIZ}$	1	Input instruction float after PSEN		38		$t_{CLCL}-25$	ns
$t_{AVIV}$	1	Address to valid instruction in		208		$5t_{CLCL}-105$	ns
$t_{PLAZ}$	1	PSEN low to address float		10		10	ns
<b>Data Memory</b>							
$t_{RLRH}$	2, 3	RD pulse width	275		$6t_{CLCL}-100$		ns
$t_{WLWH}$	2, 3	WR pulse width	275		$6t_{CLCL}-100$		ns
$t_{RLDV}$	2, 3	RD low to valid data in		148		$5t_{CLCL}-165$	ns
$t_{RHDX}$	2, 3	Data hold after RD	0		0		ns
$t_{RHDZ}$	2, 3	Data float after RD		55		$2t_{CLCL}-70$	ns
$t_{LLDZ}$	2, 3	ALE low to valid data in		350		$8t_{CLCL}-150$	ns
$t_{AVDV}$	2, 3	Address to valid data in		398		$9t_{CLCL}-165$	ns
$t_{LLWL}$	2, 3	ALE low to RD or WR low	138	238	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
$t_{AVWL}$	2, 3	Address valid to WR low or RD low	120		$4t_{CLCL}-130$		ns
$t_{QVWX}$	2, 3	Data valid to WR transition	3		$t_{CLCL}-60$		ns
$t_{WHQX}$	2, 3	Data hold after WR	13		$t_{CLCL}-50$		ns
$t_{RLAZ}$	2, 3	RD low to address float		0		0	ns
$t_{WHLH}$	2, 3	RD or WR high to ALE high	23	103	$t_{CLCL}-40$	$t_{CLCL}+40$	ns
<b>External Clock</b>							
$t_{CHCX}$	6	High time	20		20		ns
$t_{CLCX}$	6	Low time	20		20		ns
$t_{CLCH}$	6	Rise time		20		20	ns
$t_{CHCL}$	6	Fall time		20		20	ns
<b>Shift Register</b>							
$t_{XLXL}$	4	Serial port clock cycle time	750		$12t_{CLCL}$		ns
$t_{QVXH}$	4	Output data setup to clock rising edge	492		$10t_{CLCL}-133$		ns
$t_{XHQX}$	4	Output data hold after clock rising edge	8		$2t_{CLCL}-117$		ns
$t_{XHDX}$	4	Input data hold after clock rising edge	0		0		ns
$t_{XHDV}$	4	Clock rising edge to input data valid		492		$10t_{CLCL}-133$	ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.

## CMOS single-chip 8-bit microcontroller 87C528

## AC ELECTRICAL CHARACTERISTICS – I<sup>2</sup>C INTERFACE

SYMBOL	PARAMETER	INPUT	OUTPUT	I <sup>2</sup> C SPECIFICATION
<b>SCL TIMING CHARACTERISTICS</b>				
t <sub>HD; STA</sub>	START condition hold time	≥ 14 t <sub>CLCL</sub> <sup>1</sup>	Note 2	≥ 4.0μs
t <sub>LOW</sub>	SCL LOW time	≥ 16 t <sub>CLCL</sub>	Note 2	≥ 4.7μs
t <sub>HIGH</sub>	SCL HIGH time	≥ 14 t <sub>CLCL</sub> <sup>1</sup>	≥ 80 t <sub>CLCL</sub> <sup>3</sup>	≥ 4.0μs
t <sub>RC</sub>	SCL rise time	≤ 1μs <sup>1</sup>	Note 5	≤ 1.0μs
t <sub>FC</sub>	SCL fall time	≤ 0.3μs <sup>1</sup>	≤ 0.3μs <sup>6</sup>	≤ 0.3μs
<b>SDA TIMING CHARACTERISTICS</b>				
t <sub>SU; DAT1</sub>	Data set-up time	≥ 250ns	Note 2	≥ 250ns
t <sub>HD; DAT</sub>	Data hold time	≥ 0ns	Note 2	≥ 0ns
t <sub>SU; STA</sub>	Repeated START set-up time	≥ 14 t <sub>CLCL</sub> <sup>1</sup>	Note 2	≥ 4.7μs
t <sub>SU; STO</sub>	STOP condition set-up time	≥ 14 t <sub>CLCL</sub> <sup>1</sup>	Note 2	≥ 4.0μs
t <sub>BUF</sub>	Bus free time	≥ 14 t <sub>CLCL</sub> <sup>1</sup>	Note 2	≥ 4.7μs
t <sub>RD</sub>	SDA rise time	≤ 1μs <sup>4</sup>	Note 5	≤ 1.0μs
t <sub>FD</sub>	SDA fall time	≤ 0.3μs <sup>4</sup>	≤ 0.3μs <sup>6</sup>	≤ 0.3μs

**NOTES:**

1. At  $f_{CLK} = 3.5\text{MHz}$ , this evaluates to  $14 \times 286\text{ns} = 4\mu\text{s}$ , i.e., the bit-level I<sup>2</sup>C interface can respond to the I<sup>2</sup>C protocol for  $f_{CLK} \geq 3.5\text{MHz}$ .
2. This parameter is determined by the user software, it has to comply with the I<sup>2</sup>C.
3. This value gives the autolock pulse length which meets the I<sup>2</sup>C specification for the specified XTAL clock frequency range. Alternatively, the SCL pulse may be timed by software.
4. Spikes on SDA and SCL lines with a duration of less than  $4 \times f_{CLK}$  will be filtered out.
5. The rise time is determined by the external bus line capacitance and pull-up resistor, it must be  $\leq 1\mu\text{s}$ .
6. The maximum capacitance on bus lines SDA and SCL is 400pF.

## 87C528

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

C - Clock

D – Input data

H – Logic level high

- 1 – Instruction (program memory contents)

L – Logic level low, or ALE

P - PSEN

Q – Output data

R -  $\overline{RD}$  signal

t - Time

V – Valid

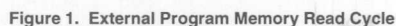
W-  $\overline{WR}$  signal

X – No longer a valid logic level

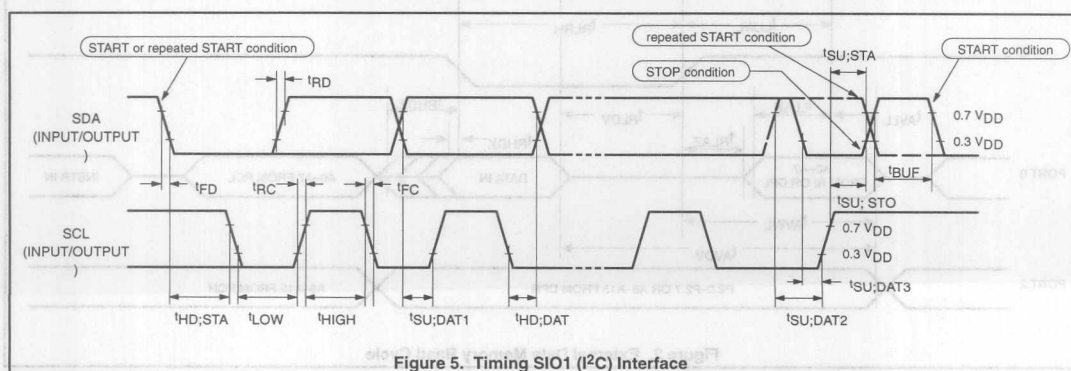
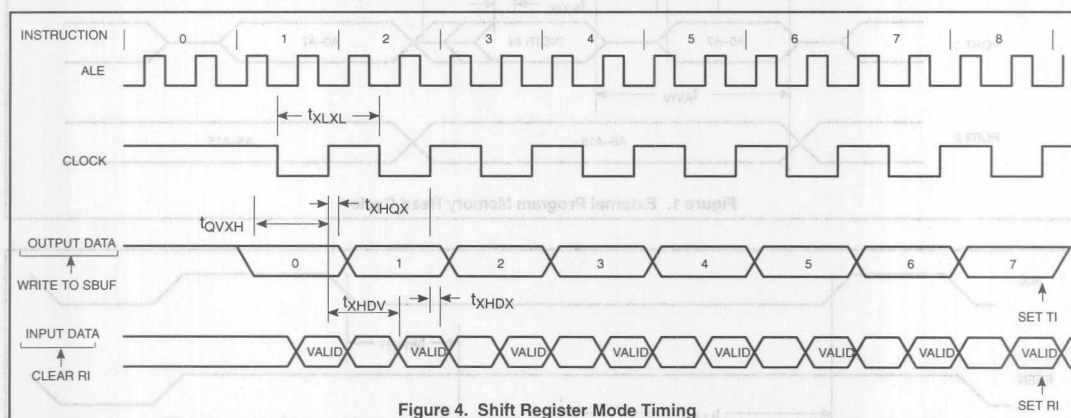
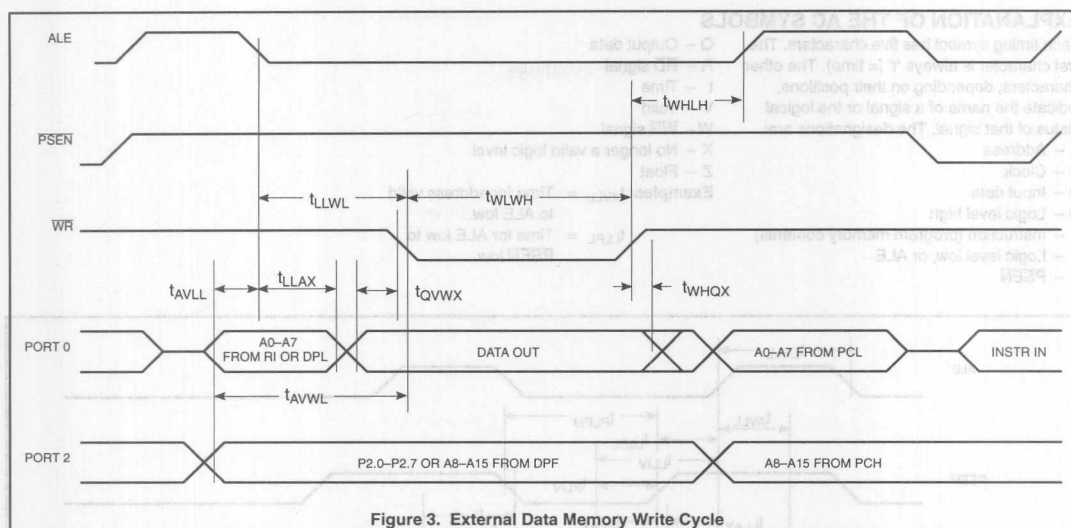
Z - Float

**Examples:**  $t_{AVLL}$  = Time for address valid to ALE low.

$t_{LLPL}$  = Time for ALE low to PSEN low.



## CMOS single-chip 8-bit microcontroller 87C528



## CMOS single-chip 8-bit microcontroller 87C528

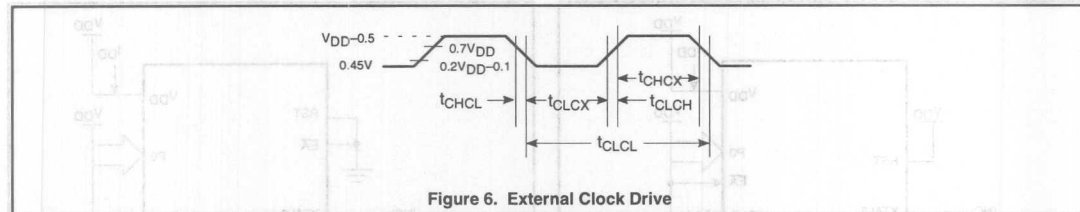


Figure 6. External Clock Drive

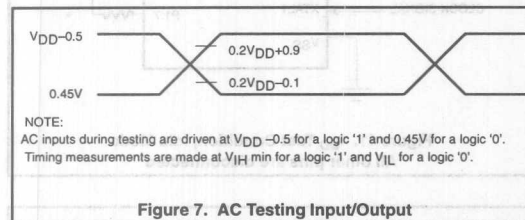


Figure 7. AC Testing Input/Output

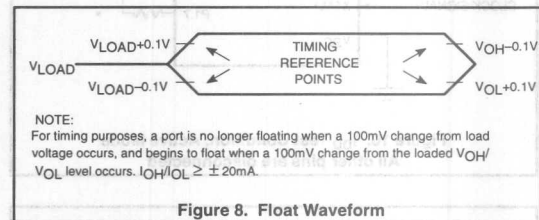
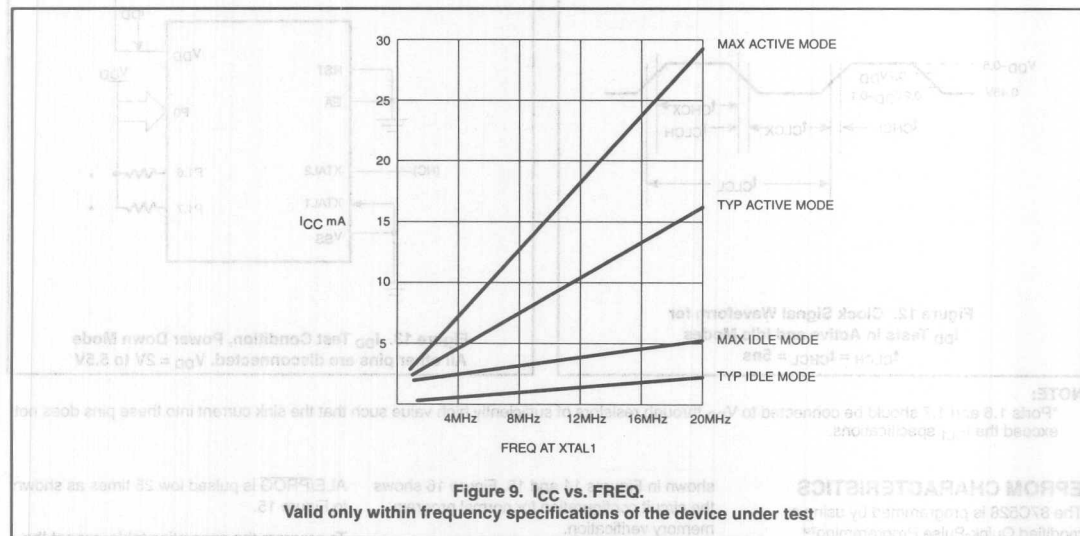


Figure 8. Float Waveform

Figure 9.  $I_{CC}$  vs. FREQ.

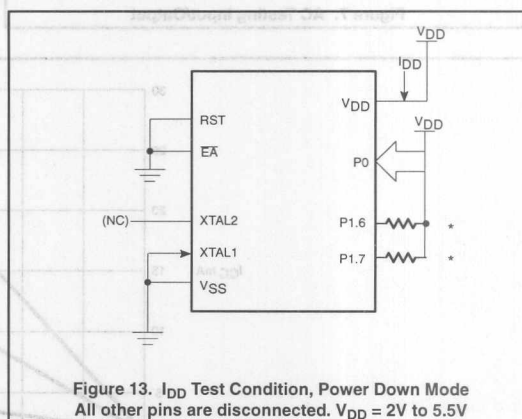
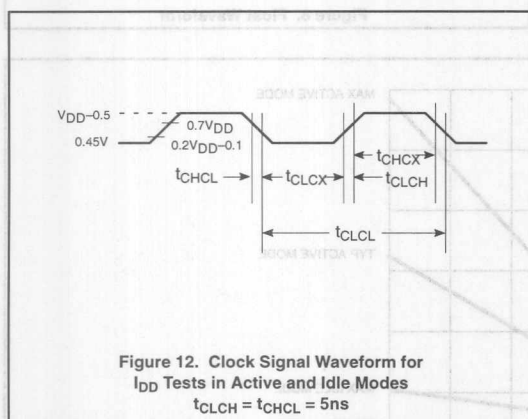
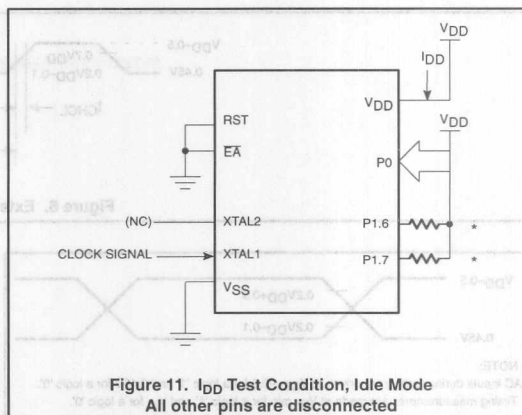
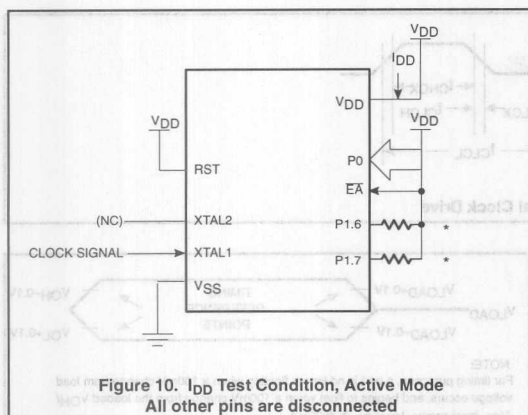
**Quick-Pulse Programming**

The setup for microcontroller data-pulse programming is shown in Figure 14. Note that the 87C528 is running with a 4 to 6 kHz oscillator. The reason the oscillator needs to be running is that the device is executing internal address and program data transfers. The address of the EPROM location to be programmed is applied to ports 2 and 3, as shown in Figure 14. The code byte to be programmed into that location is applied to port 0, P0, P2EN and pins of ports 2 and 3 specified in Table 4 are held at the Program Code Data levels indicated in Table 4. The

algorithm it differs from other methods in the values used for Vpp (programming supply voltage) and in the width and number of the ALERGO pulses. The 87C528 contains two signature bytes that can be read and used by an EPROM programming system to identify the device. The algorithm for identifying the device as an 87C528 manufactured by Philips. Table 4 shows the logic levels for reading the signature bytes. For programming the program memory, the encryption table, and the lock bits. The encryption table and waveforms for data-pulse programming are



## CMOS single-chip 8-bit microcontroller 87C528

**NOTE:**

\*Ports 1.6 and 1.7 should be connected to  $V_{DD}$  through resistors of sufficiently high value such that the sink current into these pins does not exceed the  $I_{OL1}$  specifications.

**EPROM CHARACTERISTICS**

The 87C528 is programmed by using a modified Quick-Pulse Programming™ algorithm. It differs from older methods in the value used for  $V_{PP}$  (programming supply voltage) and in the width and number of the ALE/PROG pulses.

The 87C528 contains two signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes identify the device as an 87C528 manufactured by Philips.

Table 4 shows the logic levels for reading the signature byte, and for programming the program memory, the encryption table, and the lock bits. The circuit configuration and waveforms for quick-pulse programming are

shown in Figures 14 and 15. Figure 16 shows the circuit configuration for normal program memory verification.

**Quick-Pulse Programming**

The setup for microcontroller quick-pulse programming is shown in Figure 14. Note that the 87C528 is running with a 4 to 6MHz oscillator. The reason the oscillator needs to be running is that the device is executing internal address and program data transfers.

The address of the EPROM location to be programmed is applied to ports 1, 2 and 3, as shown in Figure 14. The code byte to be programmed into that location is applied to port 0. RST, PSEN and pins of ports 2 and 3 specified in Table 4 are held at the 'Program Code Data' levels indicated in Table 4. The

ALE/PROG is pulsed low 25 times as shown in Figure 15.

To program the encryption table, repeat the 25 pulse programming sequence for addresses 0 through 3FH, using the 'Pgm Encryption Table' levels. Do not forget that after the encryption table is programmed, verification cycles will produce only encrypted data.

To program the lock bits, repeat the 25 pulse programming sequence using the 'Pgm Lock Bit' levels. After one lock bit is programmed, further programming of the code memory and encryption table is disabled. However, the other lock bit can still be programmed.

## CMOS single-chip 8-bit microcontroller 87C528

Note that the  $\overline{EA}/V_{PP}$  pin must not be allowed to go above the maximum specified  $V_{PP}$  level for any amount of time. Even a narrow glitch above that voltage can cause permanent damage to the device. The  $V_{PP}$  source should be well regulated and free of glitches and overshoot.

#### Program Verification

If lock bit 2 has not been programmed, the on-chip program memory can be read out for program verification. The address of the program memory locations to be read is applied to ports 1, 2 and 3 as shown in Figure 16. The other pins are held at the 'Verify Code Data' levels indicated in Table 4. The contents of the address location will be emitted on port 0. External pull ups are required on port 0 for this operation.

If the encryption table has been programmed, the data presented at port 0 will be the exclusive NOR of the program byte with one of the encryption bytes. The user will have to know the encryption table contents in order to

correctly decode the verification data. The encryption table itself cannot be read out.

#### Reading the Signature Bytes

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031 H, except that P3.6 and P3.7 need to be pulled to a logic low. The values are:

(030H) = 15H indicates manufactured by Philips

(031 H) = 9BH indicates 87C528

#### Program Lock Bits

The 87C528 has 3 programmable lock bits that will provide different levels of protection for the on-chip code and data (see Table 5).

Erasing the EPROM also erases the encryption array and the program lock bits, returning the part to full functionality.

#### Program/Verify Algorithms

Any algorithm in agreement with the conditions listed in Table 4, and which satisfies the timing specifications, is suitable.

#### Erase Characteristics

Erase of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent erasure. **For this and secondary effects, it is recommended that an opaque label be placed over the window.** For elevated temperature or environments where solvents are being used, apply Kapton tape Fluorglas part number 2345-5, or equivalent.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 angstroms) to an integrated dose of at least 15W-sec/cm<sup>2</sup>. Exposing the EPROM to an ultraviolet lamp of 12,000µW/cm<sup>2</sup> rating for 20 to 39 minutes, at a distance of about 1 inch, should be sufficient. Erasure leaves the array in an all 1 s state.

Table 4. EPROM Programming Modes

MODE	RST	PSEN	ALE/PROG	$\overline{EA}/V_{PP}$	P2.7	P2.6	P3.7	P3.6
Read signature	1	0	1	1	0	0	0	0
Program code data	1	0	0*	$V_{PP}$	1	0	1	1
Verify code data	1	0	1	1	0	0	1	1
Pgm encryption table	1	0	0*	$V_{PP}$	1	0	1	0
Pgm lock bit 1	1	0	0*	$V_{PP}$	1	1	1	1
Pgm lock bit 2	1	0	0*	$V_{PP}$	1	1	0	0
Pgm lock bit 3	1	0	0*	$V_{PP}$	0	1	0	1

#### NOTES:

1. '0' = Valid low for that pin, '1' = valid high for that pin.

2.  $V_{PP}$  = 12.75V  $\pm$  0.25V.

3.  $V_{CC}$  = 5V  $\pm$  10% during programming and verification.

\* ALE/PROG receives 25 programming pulses while  $V_{PP}$  is held at 12.75V. Each programming pulse is low for 100µs ( $\pm$  10µs) and high for a minimum of 10µs.

Table 5. Program Lock Bits

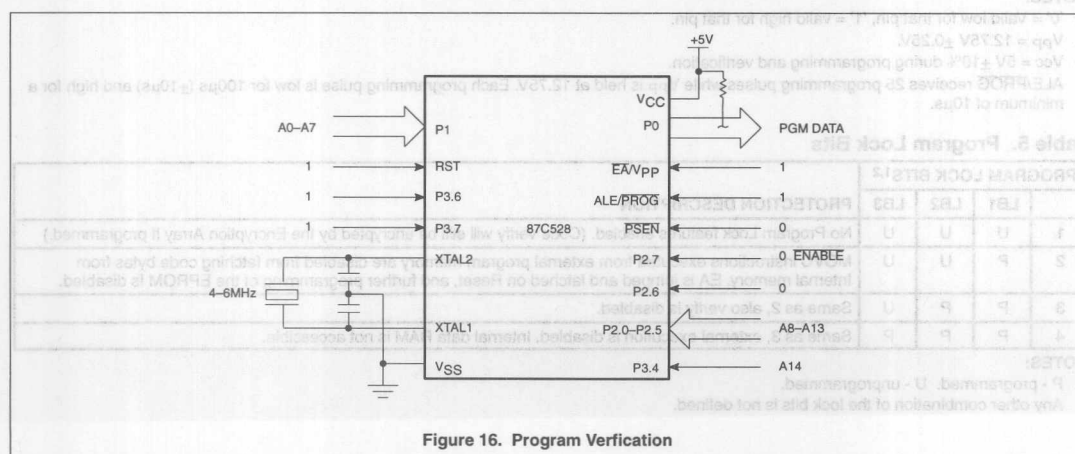
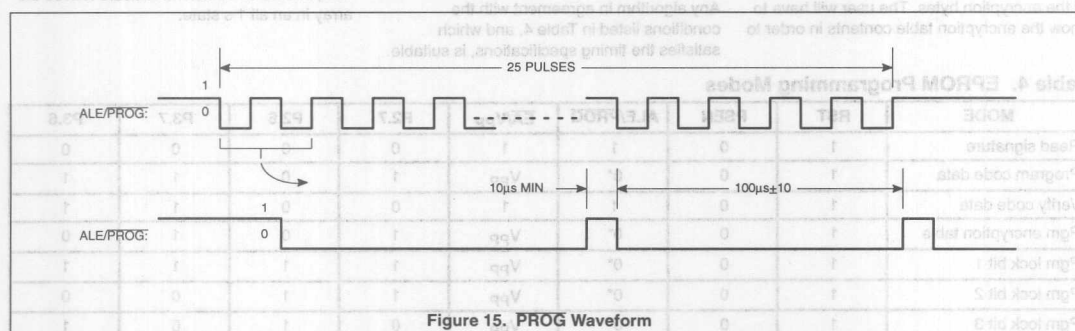
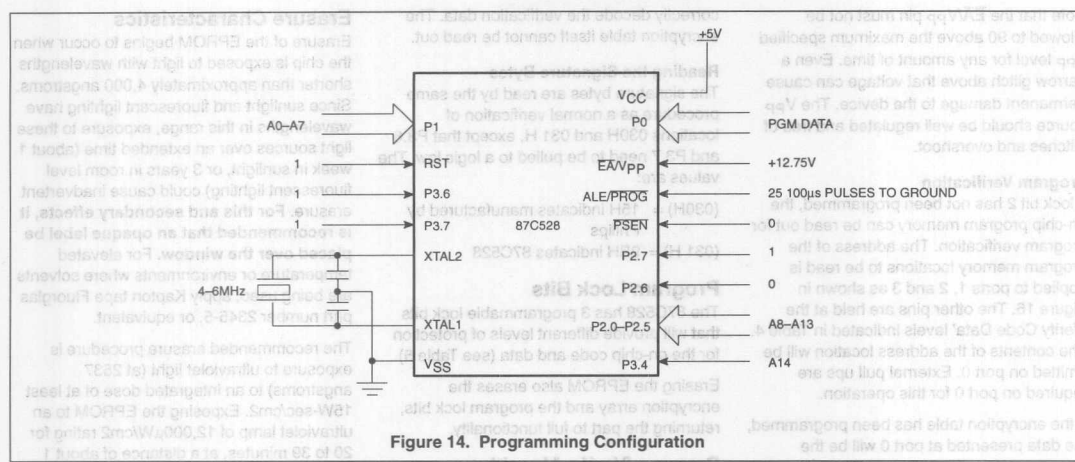
PROGRAM LOCK BITS <sup>1,2</sup>				PROTECTION DESCRIPTION
	LB1	LB2	LB3	
1	U	U	U	No Program Lock features enabled. (Code verify will still be encrypted by the Encryption Array if programmed.)
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from Internal memory, EA is jumped and latched on Reset, and further programming of the EPROM is disabled.
3	P	P	U	Same as 2, also verify is disabled.
4	P	P	P	Same as 3, external execution is disabled. Internal data RAM is not accessible.

#### NOTES:

1. P - programmed. U - unprogrammed.

2. Any other combination of the lock bits is not defined.

## CMOS single-chip 8-bit microcontroller 87C528



CMOS single-chip 8-bit microcontroller

87C528

EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

T<sub>amb</sub> = 21°C to +27°C, V<sub>cc</sub> = 5V±10%, V<sub>SS</sub> = 0V (See Figure 17)

SYMBOL	PARAMETER	MIN	MAX	UNIT
V <sub>PP</sub>	Programming supply voltage	12.5	13.0	V
I <sub>PP</sub>	Programming supply current		50	mA
1/t <sub>CLCL</sub>	Oscillator frequency	4	6	MHz
t <sub>AVGL</sub>	Address setup to PROG low	48t <sub>CLCL</sub>		
t <sub>GHAX</sub>	Address hold after PROG	48t <sub>CLCL</sub>		
t <sub>DVGL</sub>	Data setup to PROG low	48t <sub>CLCL</sub>		
t <sub>GHDX</sub>	Data hold after PROG	48t <sub>CLCL</sub>		
t <sub>EHSH</sub>	P2.7 (ENABLE) high to V <sub>PP</sub>	48t <sub>CLCL</sub>		
t <sub>SHGL</sub>	V <sub>PP</sub> setup to PROG low	10		µs
t <sub>GHSL</sub>	V <sub>PP</sub> hold after PROG	10		µs
t <sub>GLGH</sub>	PROG width	90	110	µs
t <sub>AVQV</sub>	Address to data valid		48t <sub>CLCL</sub>	
t <sub>ELQZ</sub>	ENABLE low to data valid		48t <sub>CLCL</sub>	
t <sub>EHQZ</sub>	Data float after ENABLE	0	48t <sub>CLCL</sub>	
t <sub>GHGL</sub>	PROG high to PROG low	10		µs

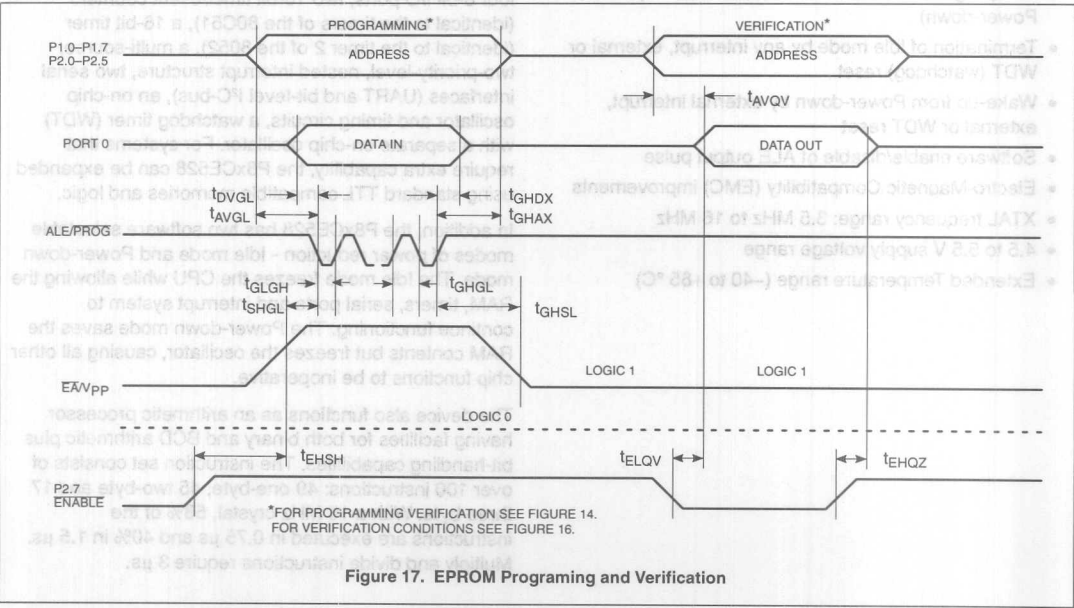


Figure 17. EPROM Programing and Verification



Purchase of Philips' I<sup>2</sup>C components conveys a license under the Philips' I<sup>2</sup>C patent to use the components in the I<sup>2</sup>C-system provided the system conforms to the I<sup>2</sup>C specifications defined by Philips.

## 8-bit microcontroller with EMC and FEEPROM

## P8xCE528

## 1 FEATURES

- 80C51 central processing unit
- 32K x 8 ROM resp. FEEPROM, expandable externally to 64 kbytes
- ROM/FEEPROM code protection
- 512 x 8 RAM, expandable externally to 64 kbytes
- Four 8-bit I/O ports
- Full-duplex UART compatible with the standard 80C51 and the 8052
- Two standard 16-bit timer/counters
- An additional 16-bit timer (functionally equivalent to the timer 2 of the 8052)
- On-chip Watchdog Timer (WDT) with an on-chip oscillator
- Bit-level I<sup>2</sup>C-bus hardware serial I/O Port
- 7-source and 7-vector interrupt structure with 2 priority levels
- Up to 3 external interrupt request inputs
- Two programmable power reduction modes (Idle and Power-down)
- Termination of Idle mode by any interrupt, external or WDT (watchdog) reset
- Wake-up from Power-down by external interrupt, external or WDT reset
- Software enable/disable of ALE output pulse
- Electro-Magnetic Compatibility (EMC) improvements
- XTAL frequency range: 3.5 MHz to 16 MHz
- 4.5 to 5.5 V supply voltage range
- Extended Temperature range (-40 to +85 °C)

## 2 GENERAL DESCRIPTION

The P83CE528; P80CE528; P89CE528 (hereafter generically referred to as P8xCE528) single-chip 8-bit microcontroller is manufactured in an advanced CMOS process and is a derivative of the PCB80C51 microcontroller family.

The P8xCE528 has the same instruction set as the 80C51. Three versions of the derivative exist:

- P83CE528: 32 kbytes mask programmable ROM
- P80CE528: ROMless version of the P83CE528
- P89CE528: 32 kbytes FEEPROM (Flash Electrically Erasable Program Memory).

This device provides architectural enhancements that make it applicable in a variety of applications in general control systems, especially in those systems which need a large ROM and RAM capacity on chip.

The P8xCE528 contains a non-volatile 32K x 8 read-only program memory (P83CE528) or FEEPROM (P89CE528), a volatile 512 x 8 read/write data memory, four 8-bit I/O ports, two 16-bit timer/event counters (identical to the timers of the 80C51), a 16-bit timer (identical to the timer 2 of the 8052), a multi-source, two-priority-level, nested interrupt structure, two serial interfaces (UART and bit-level I<sup>2</sup>C-bus), an on-chip oscillator and timing circuits, a watchdog timer (WDT) with a separate on-chip oscillator. For systems that require extra capability, the P8xCE528 can be expanded using standard TTL compatible memories and logic.

In addition, the P8xCE528 has two software selectable modes of power reduction - Idle mode and Power-down mode. The Idle mode freezes the CPU while allowing the RAM, timers, serial ports and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.

The device also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of over 100 instructions: 49 one-byte, 45 two-byte and 17 three-byte. With a 16 MHz crystal, 58% of the instructions are executed in 0.75 µs and 40% in 1.5 µs. Multiply and divide instructions require 3 µs.



## 8-bit microcontroller with EMC and FEEPROM P8xCE528

## 3 ELECTROMAGNETIC COMPATIBILITY (EMC)

Primary attention is paid to the reduction of electro-magnetic emission of the microcontroller P8xCE528. The following features reduce the electro-magnetic emission and additionally improve the electromagnetic susceptibility:

- Two supply voltage pins ( $V_{DD1,2}$ ) and four ground pins ( $V_{SS1,2,3,4}$ ) are provided on the package as follows:
  - one  $V_{DD}$  and one  $V_{SS}$  as a pair of pins placed mid-centre on one side of the package
  - a second pair of  $V_{DD}$  and  $V_{SS}$  pins placed mid-centre on the opposite side of the package
  - two more  $V_{SS}$  pins, one placed on each of the other two sides of the package.
- Separated  $V_{DD}$  pins for the internal logic and the port buffers.
- Internal decoupling capacitance improves the EMC radiation behaviour and the EMC immunity.
- External capacitors are to be located as close as possible between pins  $V_{DD1}$  and  $V_{SS1}$  as well as  $V_{DD2}$  and  $V_{SS3}$ ; ceramic chip capacitors are recommended (100 nF).

## 3.1 Recommendation on ALE

For applications that require no external memory or temporarily no external memory: the ALE output signal (pulses at a frequency of  $f_{osc}/6$ ) can be disabled under software control (bit 5 in the PCON SFR: "RFI"); if disabled, no ALE pulse will occur. ALE pin will be pulled down internally, switching an external address latch to a quiet state. The MOVX instruction will still toggle ALE as a normal MOVX. ALE will retain its normal HIGH value during Idle mode and a LOW value during Power-down mode while in the "RFI" reduction mode.

Additionally during internal access ( $\overline{EA} = 1$ ) ALE will toggle normally when the address exceeds the internal program memory size. During external access ( $\overline{EA} = 0$ ) ALE will always toggle normally, whether the flag "RFI" is set or not.

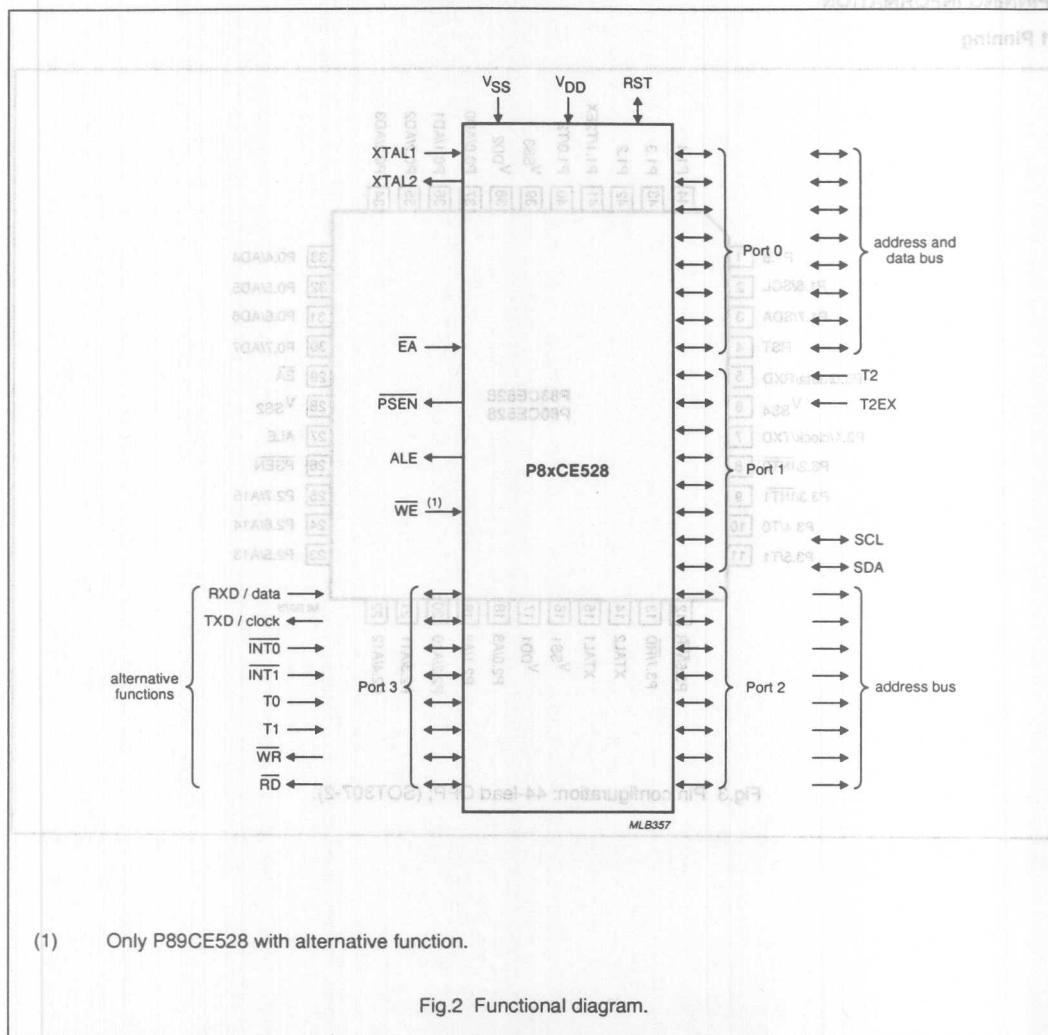
## 4 ORDERING INFORMATION

EXTENDED TYPE NUMBER	PACKAGE				TEMPERATURE RANGE (°C)	FREQUENCY (MHz)
	PINS	PIN POSITION	MATERIAL	CODE		
ROMless						
P80CE528EBB	44	QFP	plastic	SOT307-2	0 to +70	3.5 to 16
P80CE528EFB	44	QFP	plastic	SOT307-2	-40 to +85	3.5 to 16
P80CE528EBA	44	PLCC	plastic	SOT187	0 to +70	3.5 to 16
P80CE528EFA	44	PLCC	plastic	SOT187	-40 to +85	3.5 to 16
ROM						
P83CE528EBB	44	QFP	plastic	SOT307-2	0 to +70	3.5 to 16
P83CE528EFB	44	QFP	plastic	SOT307-2	-40 to +85	3.5 to 16
P83CE528EBA	44	PLCC	plastic	SOT187	0 to +70	3.5 to 16
P83CE528EFA	44	PLCC	plastic	SOT187	-40 to +85	3.5 to 16
FEEPROM						
P89CE528EBA	44	PLCC	plastic	SOT187	0 to +70	3.5 to 16
P89CE528EFA	44	PLCC	plastic	SOT187	-40 to +85	3.5 to 16

## P8XCE528



## 8-bit microcontroller with EMC and FEEPROM P8xCE528



(1) Only P89CE528 with alternative function.

Fig.2 Functional diagram.

## 8-bit microcontroller with EMC and FEEPROM P8xCE528

## 5 PINNING INFORMATION

## 5.1 Pinning

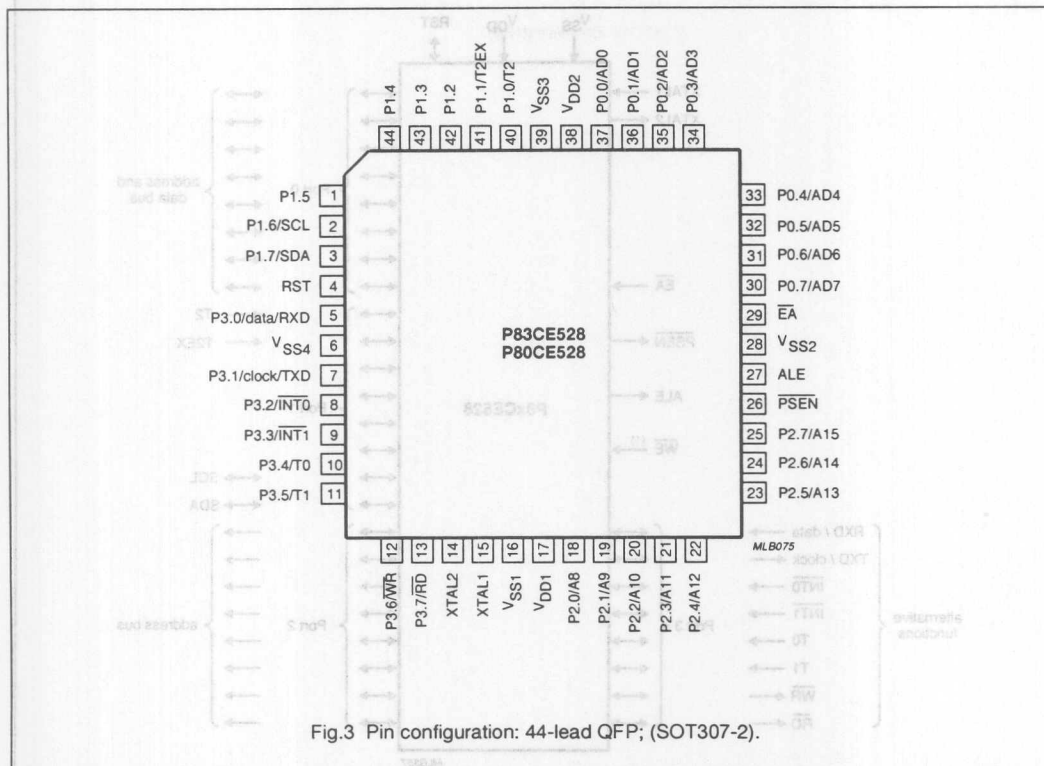
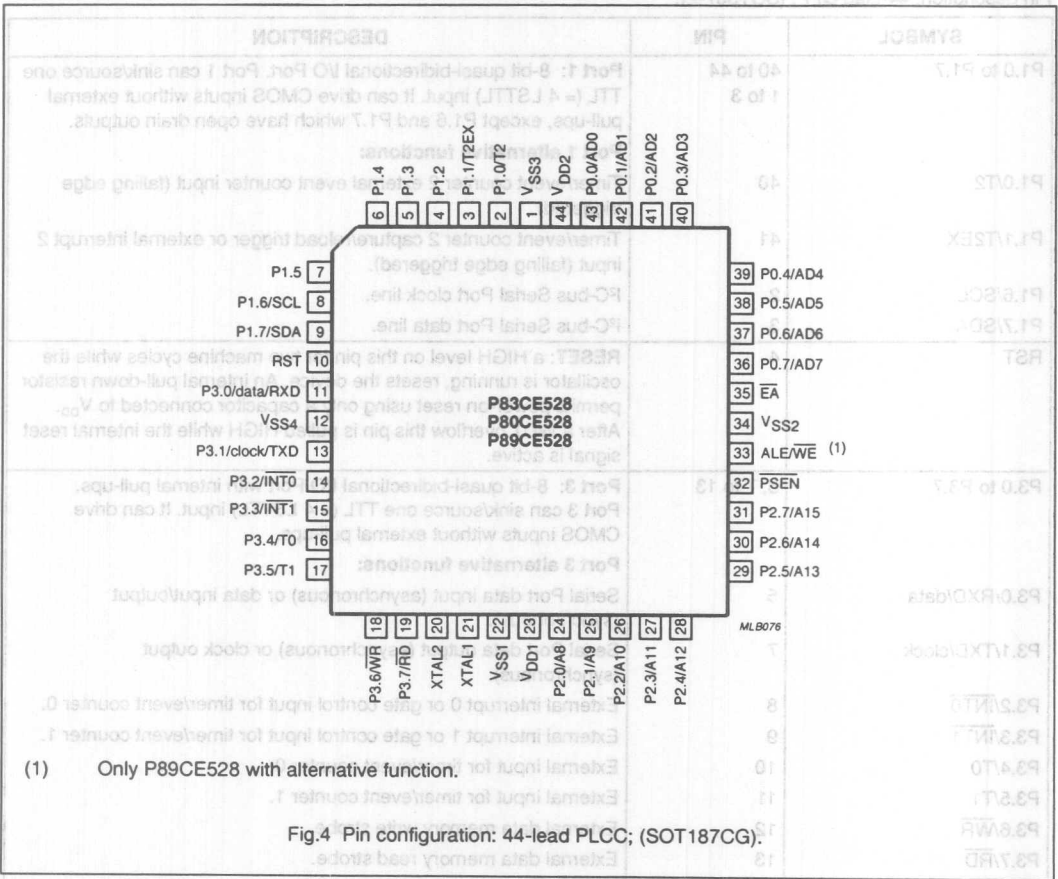


Fig.3 Pin configuration: 44-lead QFP; (SOT307-2).

8-bit microcontroller with EMC and FEEPROM P8xCE528





## 8-bit microcontroller with EMC and FEEPROM P8xCE528

Pin description: 44-lead QFP; (SOT307-2).

SYMBOL	PIN	DESCRIPTION
P1.0 to P1.7	40 to 44 1 to 3	<b>Port 1:</b> 8-bit quasi-bidirectional I/O Port. Port 1 can sink/source one TTL (= 4 LSTTL) input. It can drive CMOS inputs without external pull-ups, except P1.6 and P1.7 which have open drain outputs. <b>Port 1 alternative functions:</b>
P1.0/T2	40	Timer/event counter 2 external event counter input (falling edge triggered).
P1.1/T2EX	41	Timer/event counter 2 capture/reload trigger or external interrupt 2 input (falling edge triggered).
P1.6/SCL	2	I <sup>2</sup> C-bus Serial Port clock line.
P1.7/SDA	3	I <sup>2</sup> C-bus Serial Port data line.
RST	4	<b>RESET:</b> a HIGH level on this pin for two machine cycles while the oscillator is running, resets the device. An internal pull-down resistor permits power-on reset using only a capacitor connected to V <sub>DD</sub> . After a WDT overflow this pin is pulled HIGH while the internal reset signal is active.
P3.0 to P3.7	5, 7 to 13	<b>Port 3:</b> 8-bit quasi-bidirectional I/O Port with internal pull-ups. Port 3 can sink/source one TTL (= 4 LSTTL) input. It can drive CMOS inputs without external pull-ups. <b>Port 3 alternative functions:</b>
P3.0/RXD/data	5	Serial Port data input (asynchronous) or data input/output (synchronous).
P3.1/TXD/clock	7	Serial Port data output (asynchronous) or clock output (synchronous).
P3.2/INT0	8	External interrupt 0 or gate control input for timer/event counter 0.
P3.3/INT1	9	External interrupt 1 or gate control input for timer/event counter 1.
P3.4/T0	10	External input for timer/event counter 0.
P3.5/T1	11	External input for timer/event counter 1.
P3.6/ $\overline{WR}$	12	External data memory write strobe.
P3.7/ $\overline{RD}$	13	External data memory read strobe.
		The generation or use of a Port 3 pin as an alternative function is carried out automatically by the P8xCE528 provided the associated Special Function Register (SFR) bit is set HIGH.
XTAL2	14	<b>Crystal pin 2:</b> output of the inverting amplifier that forms the oscillator. This pin left open-circuit when an external oscillator clock is used (see Figures 15 and 16).
XTAL1	15	<b>Crystal pin 1:</b> input to the inverting amplifier that forms the oscillator, and input to the internal clock generator. Receives the external oscillator clock signal when an external oscillator is used (see Figures 15 and 16).
V <sub>SS12:3,4</sub>	16, 28, 39, 6	<b>Ground:</b> circuit ground potential. All pins must be connected.

## 8-bit microcontroller with EMC and EEPROM P8xCE528

SYMBOL	PIN	DESCRIPTION
P2.0 to P2.7	18 to 25	<b>Port 2:</b> 8-bit quasi-bidirectional I/O Port with internal pull-ups. During access to external memories (RAM/ROM) that use 16-bit addresses (MOVX @DPTR) Port 2 emits the high-order address byte (A8 to A15). Port 2 can sink/source one TTL ( $\approx 4$ LSTTL) input. It can drive CMOS inputs without external pull-ups.
PSEN	26	<b>Program Store Enable output:</b> read strobe to the external program memory via Port 0 and Port 2. It is activated twice each machine cycle during fetches from external program memory. When executing out of external program memory two activations of $\overline{\text{PSEN}}$ are skipped during each access to external data memory. PSEN is not activated (remains HIGH) during no fetches from external program memory. PSEN can sink/source 8 LSTTL inputs. It can drive CMOS inputs without external pull-ups.
ALE/ $\overline{\text{WE}}$	27	<b>Address Latch Enable output:</b> latches the LOW byte of the address during access to external memory in normal operation. It is activated every six oscillator periods except during an external data memory access. ALE/ $\overline{\text{WE}}$ can sink/source 8 LSTTL inputs. It can drive CMOS inputs without an external pull-up (note 1).
$\overline{\text{EA}}$	29	<b>External Access input:</b> when during RESET, $\overline{\text{EA}}$ is held at a TTL HIGH level, the CPU executes out of the internal program ROM, provided the program counter is less than 32768. When $\overline{\text{EA}}$ is held at a TTL LOW level during RESET, the CPU executes out of external program memory via Port 0 and Port 2. $\overline{\text{EA}}$ is not allowed to float. $\overline{\text{EA}}$ is latched during RESET and don't care after RESET.
P0.0 to P0.7	30 to 37	<b>Port 0:</b> 8-bit open drain bidirectional I/O Port. It is also the multiplexed low-order address and data bus during accesses to external memory (AD0 to AD7). During these accesses internal pull-ups are activated. Port 0 can sink/source 8 LSTTL inputs.
$V_{\text{DD1,2}}$ $V_{\text{DD1}}$ $V_{\text{DD2}}$	17 38	<b>Power supply:</b> +5 V power supply pin during normal operation and power reduction modes. Both pins must be connected. Power supply pin for ports, ALE, PSEN and on-chip oscillator. Power supply pin for internal logic.  To avoid a latch-up effect at power-on, the voltage on any pin (at any time) must not be higher than $V_{\text{DD}} + 0.5$ V or lower than $V_{\text{SS}} - 0.5$ V respectively.

## Note

1. To prohibit the toggling of ALE/ $\overline{\text{WE}}$  pin (RFI noise reduction) the bit RFI in the PCON register (PCON.5) must be set by software. This bit is cleared on RESET and can be cleared by software. When set, ALE/ $\overline{\text{WE}}$  pin will be pulled down internally, switching an external address latch to a quiet state. The MOVX instruction will still toggle ALE/ $\overline{\text{WE}}$  as a normal MOVX. ALE/ $\overline{\text{WE}}$  will retain its normal HIGH value during Idle mode and a LOW value during Power-down mode while in the "RFI" mode. Additionally during internal access ( $\overline{\text{EA}} = 1$ ) ALE/ $\overline{\text{WE}}$  will toggle normally when the address exceeds the internal program memory size. During external access ( $\overline{\text{EA}} = 0$ ) ALE/ $\overline{\text{WE}}$  will always toggle normally, whether the flag "RFI" is set or not.

## 8-bit microcontroller with EMC and FEEPROM P8xCE528

## 6 FUNCTIONAL DESCRIPTION

## 6.1 General

The P8xCE528 is a stand-alone high-performance microcontroller designed for use in real time applications such as instrumentation, industrial control, medium to high-end consumer applications and specific automotive control applications.

In addition to the 80C51 standard functions, the device provides a number of dedicated hardware functions for these applications. The P8xCE528 is a control-oriented CPU with on-chip program and data memory. It can be extended with external program memory up to 64 kbytes. It can also access up to 64 kbytes of external data memory. For systems requiring extra capability, the P8xCE528 can be expanded using standard memories and peripherals.

The P8xCE528 has two software selectable modes of reduced activity for further power reduction: Idle and Power-down. The Idle mode freezes the CPU while allowing the RAM, timers, serial ports and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator causing all other chip functions to be inoperative except the WDT if it is enabled. The Power-down mode can be terminated by an external reset, a WDT overflow, and in addition, by either of the two external interrupts.

## 6.2 Memory Organization

The central processing unit (CPU) manipulates operands in three memory spaces; these are the 64 kbyte external data memory (of which the lower 256 bytes reside in the internal AUX-RAM), 512 byte internal data memory (consisting of 256 bytes standard RAM and 256 bytes AUX-RAM) and the 64 kbyte internal and external program memory.

## 6.2.1 PROGRAM MEMORY

The program memory of the P8xCE528 consists of 32 kbyte of ROM respectively FEEPROM ("Flash Memory") on-chip, externally expandable up to 64 kbyte. If the EA pin was HIGH during RESET, the P8xCE528 executes out of the internal program memory unless the address exceeds 7FFFH. Locations 8000H through 0FFFFH are then fetched from the external program memory. If the EA pin was LOW during RESET, the P8xCE528 fetches all instructions from the external program memory. The EA input level is latched during RESET and is don't care after RESET. Figure 5 illustrates the program memory address space.

By setting a mask programmable security bit (ROM) respectively software programmable security byte (FEEPROM) the internal memory content is protected i.e. it cannot be read out by any test mode or by any instruction in the external program memory space. The MOVC instructions are the only ones which have access to program code in the internal or external program memory. The EA input is latched during RESET and is don't care after RESET. This implementation prevents reading from internal program code by switching from external program memory to internal program memory during MOVC instruction or an instruction that handles immediate data. Table 1 lists the access to the internal and external program memory by the MOVC instructions when the security feature has been activated. If the security feature is not activated, there are no restrictions for the MOVC instructions.

8-bit microcontroller with EMC and FEEPROM P8xCE528

Table 1 Internal and external program memory access with activated security feature.

INSTRUCTION	ACCESS TO INTERNAL PROGRAM MEMORY	ACCESS TO EXTERNAL PROGRAM MEMORY
MOVC in internal program memory	YES	YES
MOVC in external program memory	NO	YES

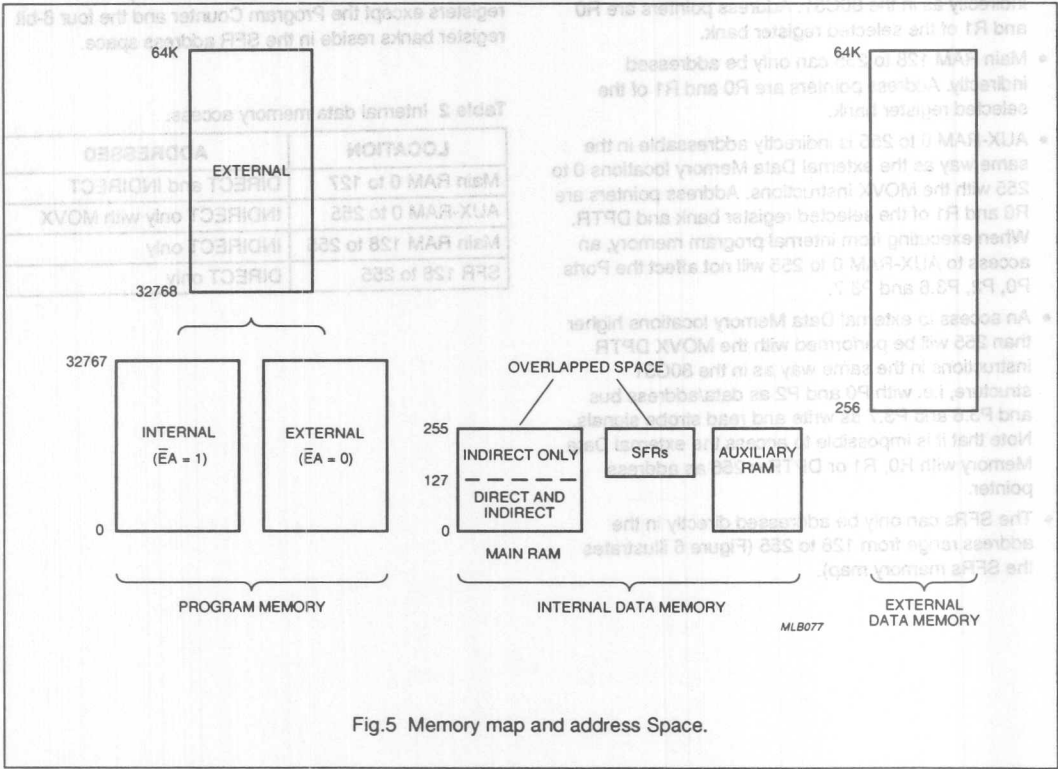


Fig.5 Memory map and address Space.

## 6.2.2 INTERNAL DATA MEMORY

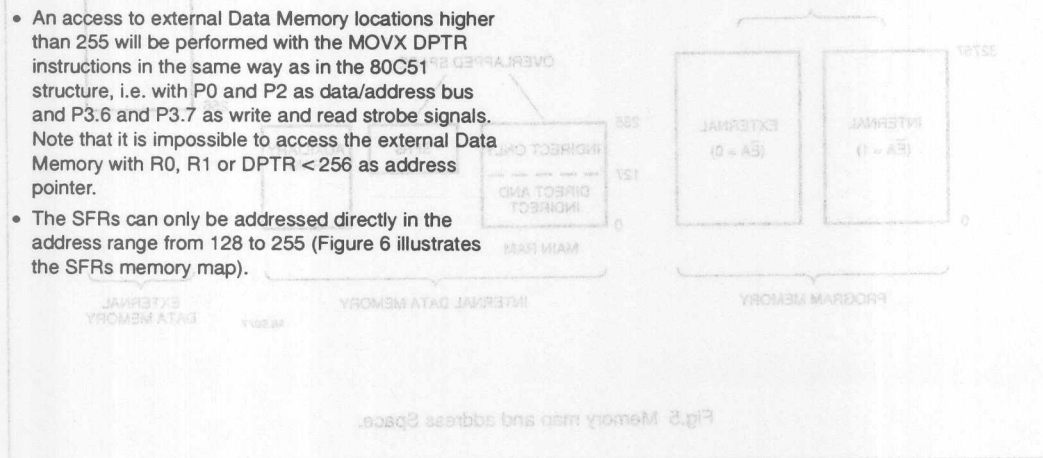
The internal data memory is divided into three physically separated parts: 256 byte of Main RAM, 256 byte of AUX-RAM, and a 128 byte special function area (SFR) see Table 2. These parts can be addressed as follows:

- Main RAM 0 to 127 can be addressed directly and indirectly as in the 80C51. Address pointers are R0 and R1 of the selected register bank.
- Main RAM 128 to 255 can only be addressed indirectly. Address pointers are R0 and R1 of the selected register bank.
- AUX-RAM 0 to 255 is indirectly addressable in the same way as the external Data Memory locations 0 to 255 with the MOVX instructions. Address pointers are R0 and R1 of the selected register bank and DPTR. When executing from internal program memory, an access to AUX-RAM 0 to 255 will not affect the Ports P0, P2, P3.6 and P3.7.
- An access to external Data Memory locations higher than 255 will be performed with the MOVX DPTR instructions in the same way as in the 80C51 structure, i.e. with P0 and P2 as data/address bus and P3.6 and P3.7 as write and read strobe signals. Note that it is impossible to access the external Data Memory with R0, R1 or DPTR < 256 as address pointer.
- The SFRs can only be addressed directly in the address range from 128 to 255 (Figure 6 illustrates the SFRs memory map).

Four 8-bit register banks occupy locations 0 through 31 in the lower RAM area. Only one of these banks may be enabled at a time. The next 16 bytes, locations 32 through 47, contain 128 directly addressable bit locations. The stack can be located anywhere in the internal 256 byte RAM. The stack depth is only limited by the available internal RAM space of 256 bytes. All registers except the Program Counter and the four 8-bit register banks reside in the SFR address space.

**Table 2** Internal data memory access.

LOCATION	ADDRESSED
Main RAM 0 to 127	DIRECT and INDIRECT
AUX-RAM 0 to 255	INDIRECT only with MOVX
Main RAM 128 to 255	INDIRECT only
SFR 128 to 255	DIRECT only





## 8-bit microcontroller with EMC and FEEPROM P8xCE528

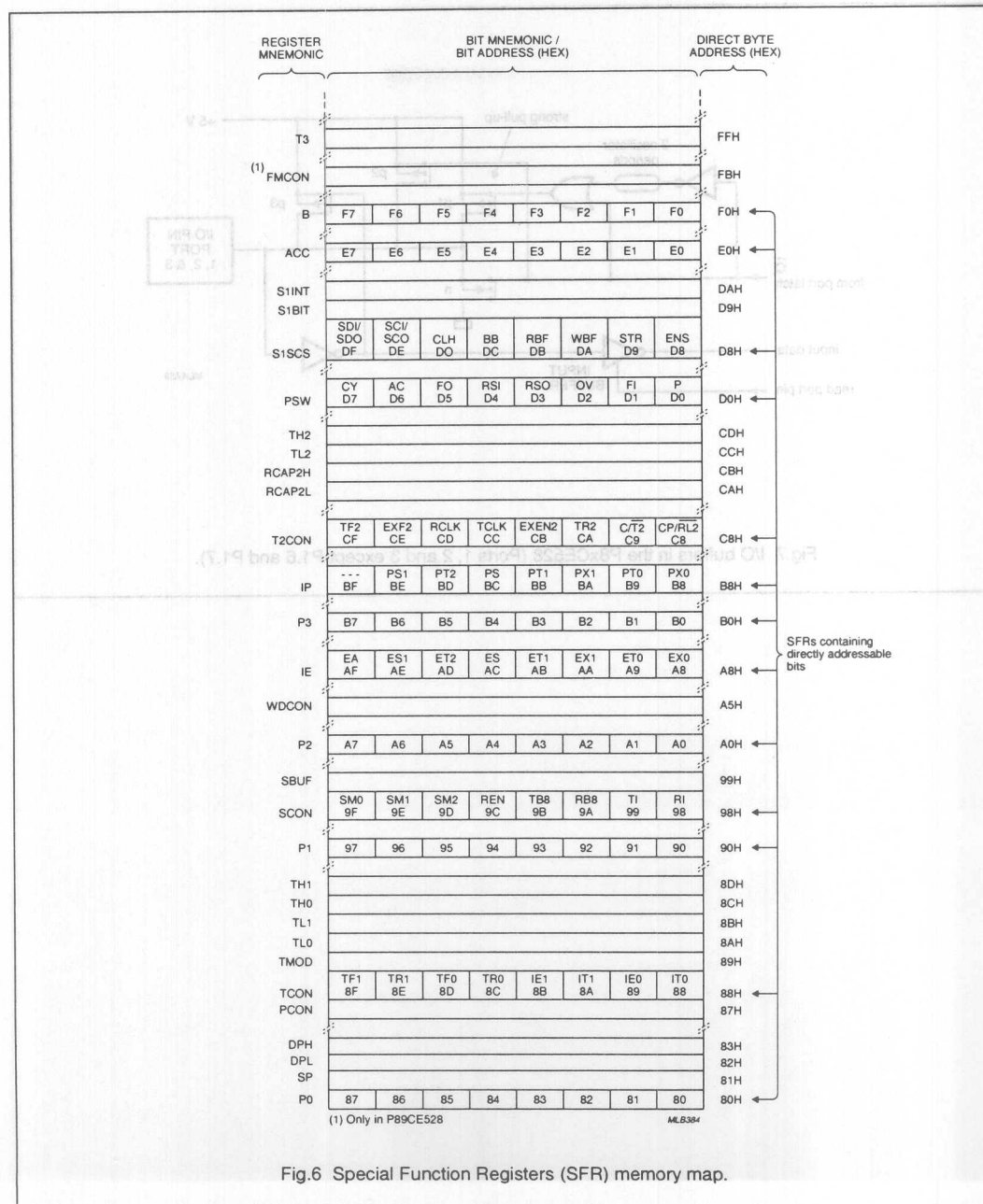
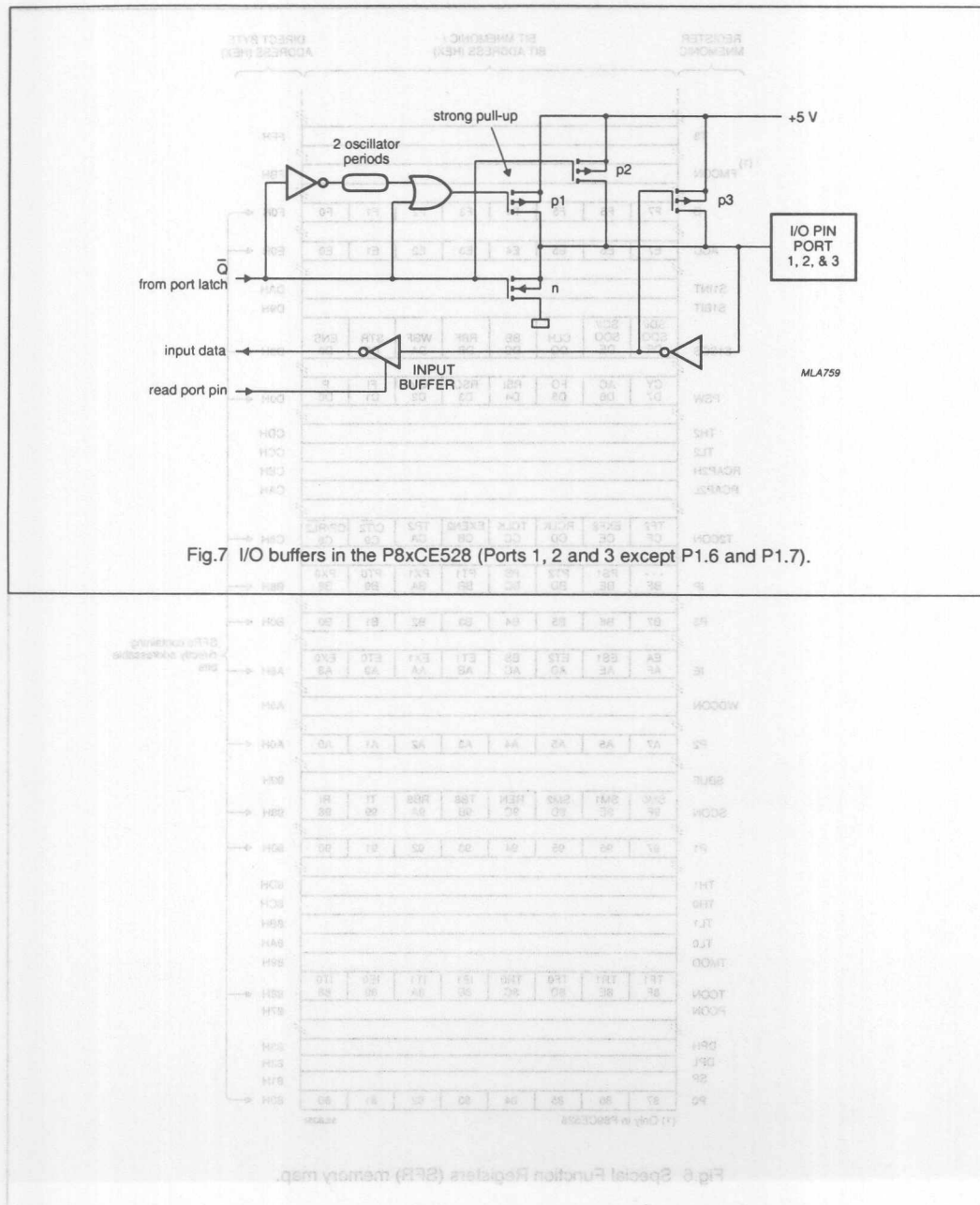


Fig.6 Special Function Registers (SFR) memory map.

## 8-bit microcontroller with EMC and FEEPROM P8xCE528



## 8-bit microcontroller with EMC and FEEPROM P8xCE528

**NOTE:** For details on the Timers, I/O, UART, WatchDog Timer, I<sup>2</sup>C interface, interrupt structure and reduced power modes, please refer to the 8XC528 data sheet in this handbook.

### 7 FLASH EEPROM

#### 7.1 General

The FEEPROM (Flash Memory) can be read and written byte-wise. Full erase, block erase, and page erase will erase 32 kbyte, 256 bytes and 32 bytes respectively. In-circuit programming and out-of-circuit programming is possible. On-chip erase and write timing generation and on-chip high voltage generation provide the following user friendly interface:

- 32 kbyte electrically erasable program memory with block and page-erase option
- Internal fixed Boot ROM
- Up to 32 kbyte external program memory in combination with the internal FEEPROM ( $\overline{EA} = 1$ )
- Up to 64 kbyte external program memory if the internal program memory is switched off ( $\overline{EA} = 0$ ).

#### 7.2 FEEPROM Features

- Read: byte-wise
- Write: byte-wise within 2.5 ms (previously erased by a page, block or full erase)
- Erase:
  - Page Erase (32 bytes) within 5 ms
  - Block Erase (256 bytes) within 5 ms
  - Full Erase (32 kbyte) within 5 ms
  - Erased bytes contain FFH
- Endurance: 100 erase and write cycles each byte at  $T_{amb} = 22^\circ\text{C}$
- Retention: 10 years
- Out-of-circuit programming: parallel programming with 87C51 compatible hardware interface to programmer
- In-circuit programming:
  - Serial programming via RS232 interface under boot ROM program control
  - Auto baud rate selection
  - Intel Hex Object File Format
- The user program can call all routines in the boot ROM for erase, write and verify of the FEEPROM.

- HIGH programming voltage generation on chip
- Zero point on-chip oscillator and timer to generate the write and erase time durations
- Programmable security for the code in the FEEPROM to prevent software piracy. The security byte is located in the highest address (7FFFH) of the FEEPROM
- Supply voltage monitoring circuit on-chip to prevent loss of information in the FEEPROM during power-on and power-off.

#### 7.3 Memory Map

Figure 8 illustrates the memory map of the user program memory and the boot ROM. They are located in the same program memory address space. Two bits UBS1 and UBS0 of the FEEPROM control special function register FMCON select between the two memory blocks.

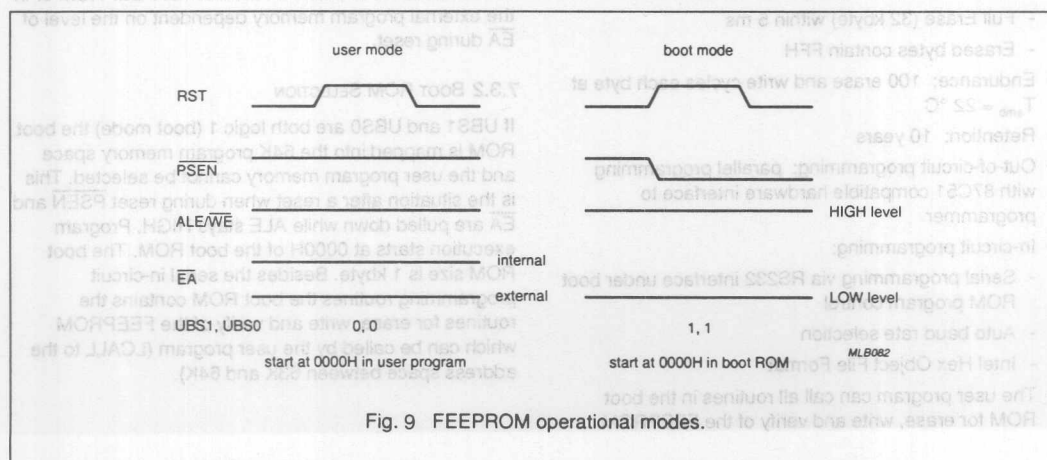
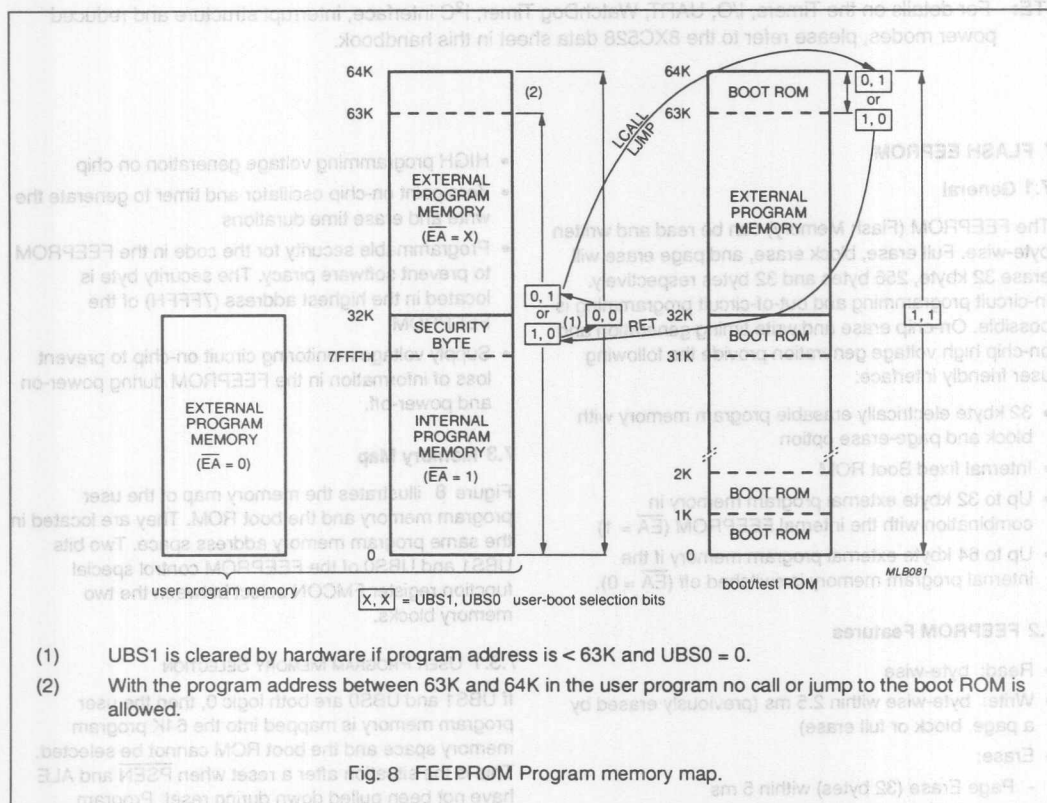
##### 7.3.1 USER PROGRAM MEMORY SELECTION

If UBS1 and UBS0 are both logic 0, then the user program memory is mapped into the 64K program memory space and the boot ROM cannot be selected. This is the situation after a reset when  $\overline{PSEN}$  and ALE have not been pulled down during reset. Program execution starts at 0000H in the internal FEEPROM or in the external program memory dependent on the level of  $\overline{EA}$  during reset.

##### 7.3.2 BOOT ROM SELECTION

If UBS1 and UBS0 are both logic 1 (boot mode) the boot ROM is mapped into the 64K program memory space and the user program memory cannot be selected. This is the situation after a reset when during reset  $\overline{PSEN}$  and ALE are pulled down while ALE stays HIGH. Program execution starts at 0000H of the boot ROM. The boot ROM size is 1 kbyte. Besides the serial in-circuit programming routines the boot ROM contains the routines for erase, write and verify of the FEEPROM which can be called by the user program (LCALL to the address space between 63K and 64K).

## 8-bit microcontroller with EMC and FEEPROM P8xCE528



8-bit microcontroller with EMC and FEEPROM P8xCE528

7.3.3 SWITCHING BETWEEN USER PROGRAM MEMORY AND BOOT ROM

Switching between user program memory (internal or external) and boot ROM is possible if UBS1 and UBS0 are 0 1 or 1 0. Then in the program memory address space between 0 and 63K the user program memory is selected and in the memory space between 63K and 64K the boot ROM is selected.

To switch from user program memory to boot ROM first UBS0 must be set (UBS1 stays 0) and a jump or call instruction to a location > 63K must be executed. At the moment of crossing the 63K address border by a call or jump instruction the switching between user and boot memory is performed without timing problems.

Switching from boot ROM to user program memory is performed as follows: UBS1 is set, UBS0 is cleared and a return instruction to a location < 63K is executed. At the moment of crossing the 63K address border by the return instruction the switching from boot ROM to user memory (internal or external) is performed. After crossing the 63K address border UBS1 is immediately cleared by hardware and the total 64K memory space is mapped as user program memory. By clearing UBS1 by hardware, no special requirements to the user program are necessary to do that after a boot routine or erase or write routine.

A small restriction for memory switching is that no memory switching is allowed from or to the address space between 63K and 64K of the user program memory because the UBS bits must stay logic 0 in this range and no 63K address crossing would take place. This restriction can be avoided if the memory switching is always done by a subroutine in the address range between 0 and 63K.

7.3.4 DESCRIPTION

The user program code in the FEEPROM is executed as in the standard 80C51 microcontroller. Erase and write cycles in the FEEPROM are always performed under control of the boot program in the boot ROM in the address space between 63K and 64K. Address and data parameters are passed via DPTR and accumulator A respectively. During an erase or write cycle in the FEEPROM no other access or program execution in the FEEPROM is possible. All interrupts must be disabled when the user program calls a user routine in the boot ROM.

The boot routine for serial programming takes care of addressing, data transfer, blank-check, verify, high voltage control, error message, interrupt disabling/enabling and return to the user program memory. It also contains the serial communication routine.

The FEEPROM control register FMCON is a special function register (SFR). It contains the control bits for verify, write, erase and boot ROM switching.

	UBS0	UBS1
User memory mapped from 0 to 63K	0	0
User memory mapped from 63K to 64K	1	0
User memory mapped from 0 to 63K in this user address range	0	1
Boot ROM mapped from 63K to 64K	1	1

	FCB0	FCB1	FCB2	FCB3
Value after reset	0	0	0	0
Byte write or byte read (verify)	1	0	1	0
Page erase (32 byte boundaries)	0	0	1	1
Block erase (256 byte boundaries)	1	1	0	0
Full erase (32 kbytes)	0	1	0	1



## 8-bit microcontroller with EMC and FEEPROM P8xCE528

## 7.3.5 FMCON: CONTROL REGISTER

FMCON SFR (FBH).

bit-7	6	5	4	3	2	1	0
UBS1	UBS0	HV	—	FCB3	FCB2	FCB1	FCB0

Table 3 Description of FMCON bits.

MNEMONIC	BIT	FUNCTION
UBS1	FMCON.7	User/boot selection bit.
UBS0	FMCON.6	User/boot selection bit.
HV	FMCON.5	HIGH voltage indication bit. Read only, is logic 1 as long as the HIGH voltage for an erase or write operation is present.
—	FMCON.4	Reserved for future use. A write operation must write 0 to this bit location.
FCB3	FMCON.3	Function code bit.
FCB2	FMCON.2	Function code bit.
FCB1	FMCON.1	Function code bit.
FCB0	FMCON.0	Function code bit.

Table 4 FMCON UBS1 and UBS0 bit modes.

UBS1	UBS0	MODE
0	0	User memory mapped from 0 to 64K.
0	1	User memory mapped from 0 to 63K. Boot ROM mapped from 63K to 64K.
1	0	User memory mapped from 0 to 63K, but UBS1 bit cleared by hardware in this user address range. Boot ROM mapped from 63K to 64K.
1	1	Boot ROM mapped from 0 to 64K.

Table 5 FMCON FCB-bit function codes.

FCB3	FCB2	FCB1	FCB0	FUNCTION CODE
0	0	0	0	Value after reset.
0	1	0	1	Byte write or byte read (verify).
1	1	0	0	Page erase (32 byte boundaries).
0	0	1	1	Block erase (256 byte boundaries).
1	0	1	0	Full erase (32 kbyte).

## 8-bit microcontroller with EMC and FEEPROM P8xCE528

The four FCB bits are write protected if the security feature is activated. Then only instructions in the internal program memory (FEEPROM) are able to write FCB3-0. Boot ROM and external program memory instructions cannot change FCB 3 to 0 except the full erase code can be loaded.

The duration of a write or erase operation is determined by the FEEPROM timer. This timer includes a zero point RC oscillator and can not be controlled by software.

For calling a user routine in the boot ROM first all interrupts must be disabled and the DPTR and A have to be loaded with the desired values. After setting UBS0 = 1 and UBS1 = 0 and selecting the function via FCB-bits the respective user routine has to be called.

Table 6 lists the boot ROM user routines, which can be called by the user program. The contents of FMCON, A and DPTR before calling is described by (IN) and by (OUT) after return. The boot ROM user routines do not change other registers or data memory.

**Table 6** Boot ROM user routines (notes 1 and 2).

BOOT ROM ROUTINE	CALL ADDRESS	FMCON (IN)	FMCON (OUT)	A (IN)	A (OUT)	DPTR (IN)	DPTR (OUT)
BYTE_READ	FFBAH	45H	15H	XXH	BYTE	BYTE ADDRESS	BYTE ADDRESS
BYTE_WRITE	FFADH	45H	15H	BYTE	BYTE (V)	BYTE ADDRESS	BYTE ADDRESS
PAGE_ERASE	FFAAH	4CH	1CH	XXH	08H	PAGE ADDRESS (note 3)	PAGE ADDRESS (note 4)
BLOCK_ERASE	FFA5H	43H	13H	XXH	02H	BLOCK ADDRESS (note 5)	BLOCK ADDRESS (note 6)
FULL_ERASE	FFA0H	4AH	1AH	XXH	0AH	XXXXH	0018H

### Notes

1. X = don't care or not defined.
2. V = verified byte (read back).
3. = 5 LSB's of DPTR are don't care.
4. = 5 LSB's of DPTR are 0.
5. = 8 LSB's of DPTR are don't care.
6. = 8 LSB's of DPTR contain 08H.

### 7.3.6 USER SOFTWARE EXAMPLE OF FEEPROM PAGE ERASE ROUTINE

Example of user software (internal or external) that calls the page erase routine in the boot ROM to erase a page in the FEEPROM (32 bytes) starting at address location 1260H:

CLR EA; disable all interrupts

MOV DPTR, #1260H; load page address

MOV FMCON, #4CH; load page erase code

LCALL 0FFAAH; call page erase routine in boot ROM (5 ms)

SETB EA; enable interrupts again

### 7.3.7 USER SOFTWARE EXAMPLE OF FEEPROM BYTE WRITE ROUTINE

Example of user software (internal or external) that calls the byte write routine in the boot ROM to write the content of R5 into the FEEPROM address location 1263H:

CLR EA; disable all interrupts

MOV DPTR, #1263H; load byte address

MOV A, R5; load byte to be written

MOV FMCON, #45H; load byte write code

LCALL 0FFADH; call byte write routine in boot ROM (2.5 ms)

SETB EA; enable interrupts again

XRL A, R5; compare the "read back" byte

JNZ . . .; jump if verify error

### 7.4 Security Feature

The security feature (security mode) protects against software piracy by limiting the read/write access to the contents of the FEEPROM. The security feature is activated or de-activated by a security code written into a security byte. If the security feature is de-activated, then there are no access restrictions to the FEEPROM.

The security byte is located in the highest address location 7FFFH of the FEEPROM. The security byte is functionally the same as all other bytes of the FEEPROM.

The security byte should be written 01010000B to activate and 00H or FFH to de-activate the security feature. The security code is chosen in such a way that single bit failures will not change the security mode.

If the security feature is activated then the external program memory has no access to the FEEPROM with the MOVX instructions. Also bits FCB 3 to 0 of FMCON cannot be written by external program code or boot ROM code. This prevents in-circuit programming and verification. Only the full erase code can be written to FCB 0 to 3 of FMCON. At the end of a full erase operation the security feature is de-activated. Parallel programming and verify is also inhibited if the security feature is activated, only a full erase is possible.

Note that the security mode does not change immediately when the security code is written into the security byte 7FFFH, but only after a reset or power-on. This allows the verification of the loaded code in the FEEPROM, including the security byte.

## 8-bit microcontroller with EMC and FEEPROM P8xCE528

### 7.5 Parallel Programming

Unlike standard EPROM programming, no HIGH programming supply voltage must be applied to the  $\overline{EA}$  pin and only one programming pulse must be applied to the ALE/WE pin.

The parallel programming mode is entered with the steady signals  $RST = 1$ ,  $\overline{PSEN} = 0$ ,  $\overline{EA} = 1$ . The XTAL clock must have a frequency between 4 MHz and 6 MHz. Table 7 shows the logic levels for programming, erasing, verifying and read signature.

**Table 7** Parallel programming modes.  
(notes 1 and 2)

MODE	ALE/WE	P2.7	P2.6	P3.7	P3.6
Full Erase	note 3	1	1	0	1
Program FEEPROM	note 4	1	0	1	1
Verify FEEPROM	1	0	0	1	1
Read signature	1	0	0	0	0

#### Notes

1. ALE/WE Write enable signal (program/erase), active LOW.
2. P2.7, P2.6, P3.7, P3.6 Output enable signals for verify/read modes, active LOW.
3. One 5 ms LOW pulse.
4. One 2.5 ms LOW pulse.

**Table 8** Data and address bits.

P0.0 to P0.7	D0 to D7	Program data input/verify or read data output.
P1.0 to P1.7	A0 to A7	Input low order address bits.
P2.0 to P2.5; P3.4	A8 to A14	Input high order address bits.

#### 7.5.1 89CE528 SIGNATURE BYTES

The 89CE528 contains two signature bytes that can be read and used by an EPROM programming system to identify the device. These bytes are read by the same procedure as for a normal verification of locations 30H and 31H, except that P3.6 and P3.7 need to be pulled to LOW. The values are given in Table 9 :

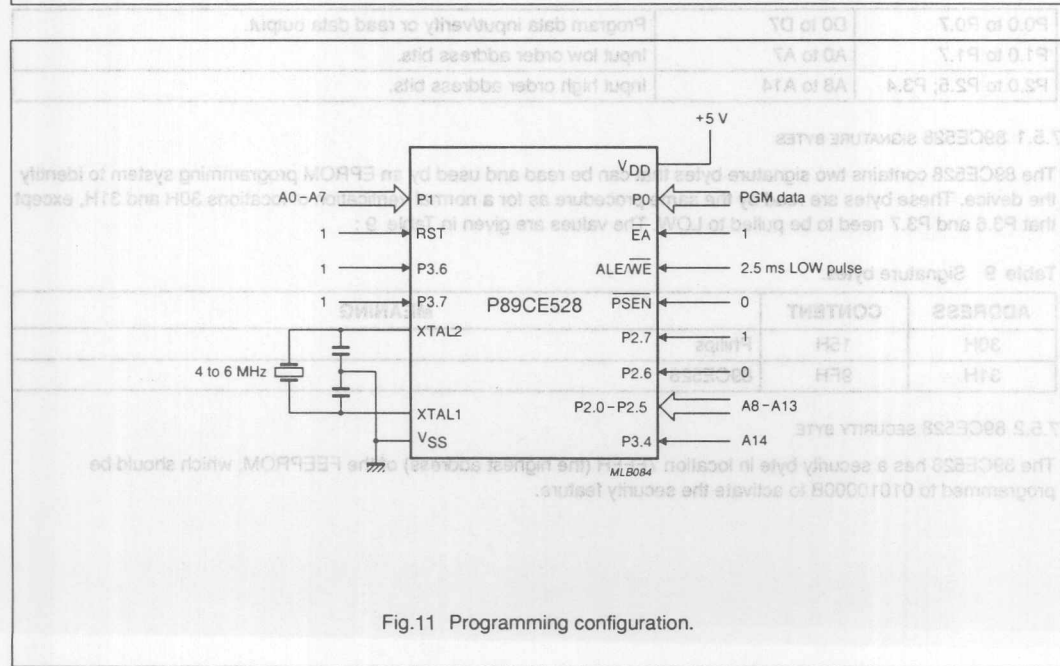
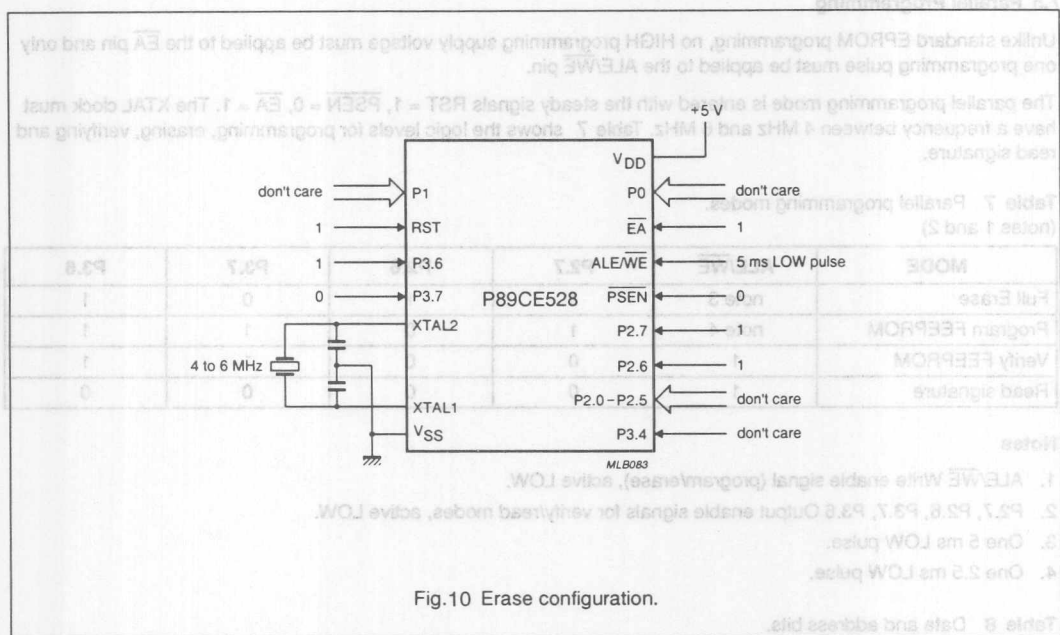
**Table 9** Signature bytes.

ADDRESS	CONTENT	MEANING
30H	15H	Philips
31H	9FH	89CE528

#### 7.5.2 89CE528 SECURITY BYTE

The 89CE528 has a security byte in location 7FFFH (the highest address) of the FEEPROM, which should be programmed to 0101000B to activate the security feature.

# 8-bit microcontroller with EMC and FEEPROM P8xCE528





## 8-bit microcontroller with EMC and EEPROM P89CE528

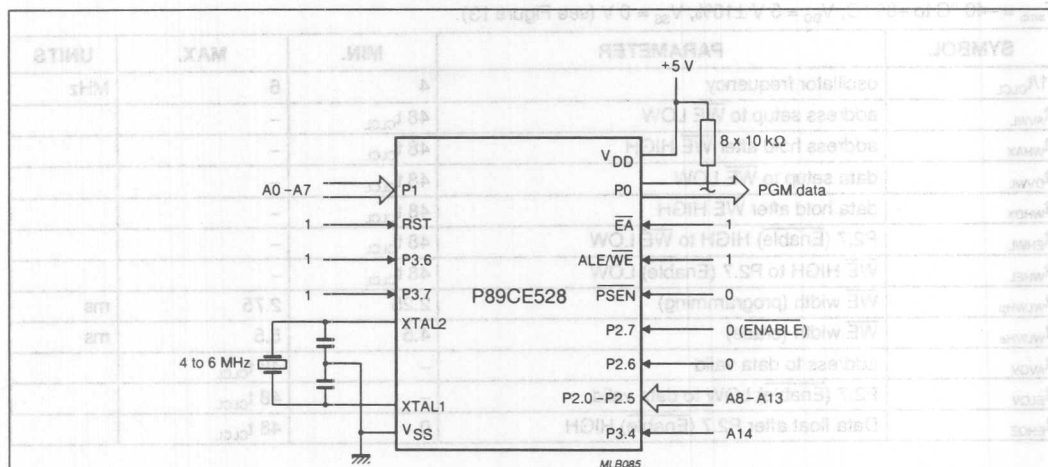
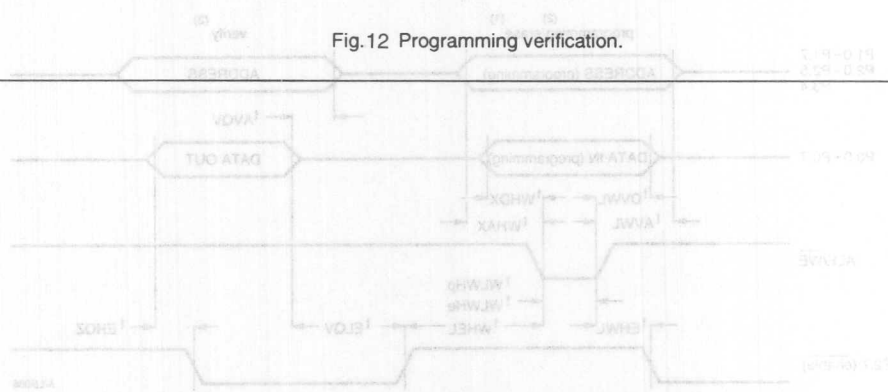


Fig.12 Programming verification.



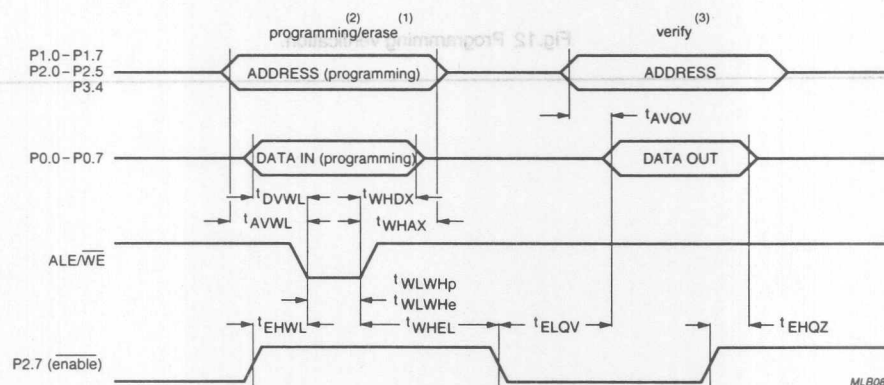
- (1) For erase conditions see Figure 10.  
 (2) For program conditions see Figure 11.  
 (3) For verify conditions see Figure 12.

Fig.13 EEPROM Programming and Verification waveforms.

## 8 FEEPROM PROGRAMMING/ERASE AND VERIFICATION CHARACTERISTICS

$T_{amb} = -40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ;  $V_{DD} = 5\text{ V} \pm 10\%$ ;  $V_{SS} = 0\text{ V}$  (see Figure 13).

SYMBOL	PARAMETER	MIN.	MAX.	UNITS
$1/t_{CLCL}$	oscillator frequency	4	6	MHz
$t_{AVWL}$	address setup to $\overline{\text{WE}}$ LOW	$48 t_{CLCL}$	—	
$t_{WHAX}$	address hold after $\overline{\text{WE}}$ HIGH	$48 t_{CLCL}$	—	
$t_{DVWL}$	data setup to $\overline{\text{WE}}$ LOW	$48 t_{CLCL}$	—	
$t_{WHDX}$	data hold after $\overline{\text{WE}}$ HIGH	$48 t_{CLCL}$	—	
$t_{EHWL}$	P2.7 ( $\overline{\text{Enable}}$ ) HIGH to $\overline{\text{WE}}$ LOW	$48 t_{CLCL}$	—	
$t_{WHEL}$	$\overline{\text{WE}}$ HIGH to P2.7 ( $\overline{\text{Enable}}$ ) LOW	$48 t_{CLCL}$	—	
$t_{WLWHp}$	$\overline{\text{WE}}$ width (programming)	2.25	2.75	ms
$t_{WLWHe}$	$\overline{\text{WE}}$ width (erase)	4.5	5.5	ms
$t_{AVQV}$	address to data valid	—	$48 t_{CLCL}$	
$t_{ELQV}$	P2.7 ( $\overline{\text{Enable}}$ ) LOW to data valid	—	$48 t_{CLCL}$	
$t_{EHQZ}$	Data float after P2.7 ( $\overline{\text{Enable}}$ ) HIGH	0	$48 t_{CLCL}$	



- (1) For erase conditions see Figure 10.
- (2) For program conditions see Figure 11.
- (3) For verify conditions see Figure 12.

Fig.13 FEEPROM Program/Erase and Verification waveforms.

## 8-bit microcontroller with EMC and FEEPROM P8xCE528

## 8.1 Serial Programming of FEEPROM

Serial in-circuit programming (Boot Mode) is entered if during and after RESET  $\overline{\text{PSEN}}$  and  $\overline{\text{EA}}$  are pulled down ( $\overline{\text{PSEN}}$  via a resistor of 2 k $\Omega$  to  $V_{\text{SS}}$ ). The two UBS bits are set to logic 1 by hardware and program execution starts at 0000H of the Boot ROM. P3.0 (RXD) and P3.1 (TXD) form the serial RS232 interface.

The receive and transmit channel have the same baud rate. The boot routine is totally in software without the use of timers or UART. It inputs the Intel Hex Object Format. The baud rate will be selected automatically after reception of the first character (:) of the object file. No other characters are allowed to precede the first (:) character. Programming is only started if the first received record has the right type indication (TT). If the

security feature is activated then the programming starts with a Full Erase, otherwise only the addressed page(s) will be erased and the not altered bytes are rewritten. During the erase or write operation the next string of bytes can be received. Xon and Xoff handshake codes are used to control the serial transfer.

At the end of the programming a message that indicates a successful or not successful programming will be returned over the RS232 interface channel. If the programming was successful then the user program can be started up at 0000H in the FEEPROM by a reset for user mode ( $\overline{\text{EA}} = \text{HIGH}$ ,  $\overline{\text{PSEN}}$  not affected). If the programming was not successful the boot program halts and a retry can be started by a reset for the Boot Mode.

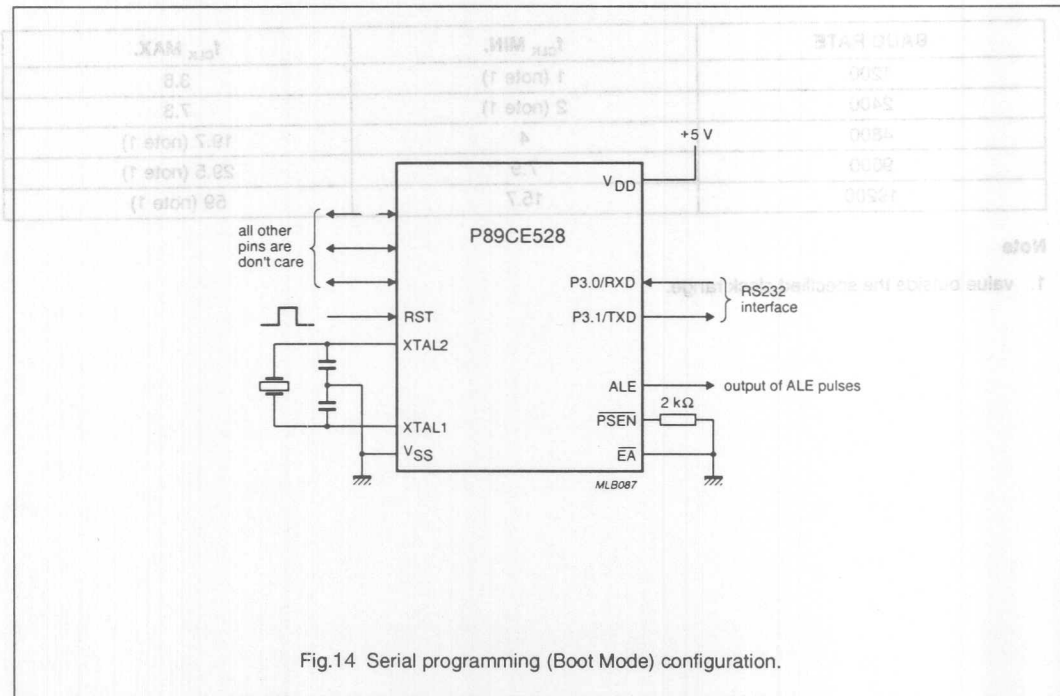


Fig.14 Serial programming (Boot Mode) configuration.

## 8-bit microcontroller with EMC and FEEPROM P8xCE528

## 8.2 Boot Routine

The boot routine transmits the next "one ASCII character" messages via the RS232 interface:

- "." after each record type TT = 00H indication in the HEX file
- "X" checksum error of a record in the HEX file detected
- "Y" wrong record type received
- "Z" buffer overflow error (Check Xon/Xoff of terminal)
- "R" verification error (of last written byte)
- "V" end record received and programming of FEEPROM was successful.

No messages are transmitted if the baud rate of the first character (:) can not be detected.

The boot routine can also be started by the internal or external user program (LJMP FC07H). FMCON must be loaded previously with 40H. Interrupt registers, stack pointer, Timer 0, UART, P3.0 and P3.1 must be in the reset state. EA and PSEN must not be affected. A reset is needed to restart the user program after programming.

Note that the boot routines can (re)program any number of bytes from 1 byte to 32 kbytes, independent in which order or at which location. The next baud rates will be detected automatically within the specified  $\mu\text{C}$  clock range in MHz.

BAUD RATE	$f_{\text{CLK}}$ MIN.	$f_{\text{CLK}}$ MAX.
1200	1 (note 1)	3.6
2400	2 (note 1)	7.3
4800	4	19.7 (note 1)
9600	7.9	29.5 (note 1)
19200	15.7	59 (note 1)

## Note

- value outside the specified clock range.

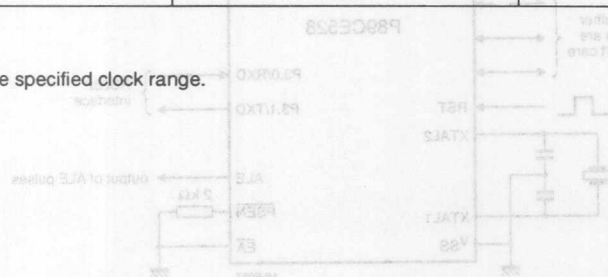


Fig. 14 Serial programming (Boot Mode) configuration

## 8-bit microcontroller with EMC and FEEPROM P8xCE528

### 8.3 Boot Routine Definitions

In the following description, each letter corresponds to one hexadecimal digit in ASCII representation (the digits 0 through 9, and the letters A through F).

Construction of data records (using the notation defined in Table 10) is as follows:

:BCAAAATTNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN  
NNHHCC

The last record in a file is the end record and contains no data. Usually the end record will appear as shown in the first example below. However, in some cases a 16 bit checksum of all the data bytes in the entire file may be inserted in the address field of the end record. This checksum would correspond to one generated by an EPROM programmer during file load, and its inclusion does not violate the rules for this format. This is shown in the second example.

Successive HEX (H) records need not appear in sequential address order. For instance, a record for address 0000H might appear after a record for address 7FE0H. All of the bytes in a single record, however, must be in sequence. Any characters that appear outside of a record (i.e. after a checksum, but before the next ":") should be ignored, if present.

An example of a valid HEX file follows:

```
:10010000C2F0E53030E74F404D2F08531F030F786
:100110000763F0FF05F0B2F0A430F00A63F0FFF4DB
:0C0120002401500205F085F032F5332276
:00000001FF
```

SYMBOL	PARAMETER	CONDITIONS	MIN	MAX
00B12C0122				:00000001FF

**Table 10** Boot routine definitions.

MNEMONIC	FUNCTION
:	Record start character.
BC	Byte count. The hexadecimal number of data bytes in the record. This may theoretically be any number from 0 to 255 prefer to deal with 16 data bytes per record (as shown in the example above).
AAAA	Load address in hexadecimal of first data byte in this record.
TT	Record type. The record type is 00 for data records and 01 for the end record.
HH	One hexadecimal data byte.
CC	Record checksum. This is the two's complement of the summation of all of the bytes in the record from the byte count through the last data byte. While summation is calculated, it is always truncated to a one byte result. Thus, if all of the bytes in the record are summed, including the checksum itself, the result will always be 00 if the record is valid.



## 8-bit microcontroller with EMC and FEEPROM P8xCE528

## 9 LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134).

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
$V_{DD}$	supply voltage range	—	−0.5	+6.5	V
$V_I$	all input voltages	—	−0.5	$V_{DD} + 0.5$	V
$P_{tot}$	total power dissipation	note 1	—	1	W
$T_{stg}$	storage temperature range		−65	+150	°C
$T_{amb}$	operating ambient temperature range:				
	version EBx		0	+70	°C
	version EFx		−40	+85	°C

## Note

1. This value is based on the maximum allowable die temperature and the thermal resistance of the package, not on the device power consumption.

## 10 DC CHARACTERISTICS (EBx)

$V_{DD} = 5\text{ V} \pm 10\%$ ;  $V_{SS} = 0\text{ V}$ ;  $T_{amb} = 0\text{ to }+70^\circ\text{C}$ . All voltages with respect to  $V_{SS}$  unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
<b>Supply</b>					
$V_{DD}$	supply voltage range		4.5	5.5	V
<b>Supply current</b>					
$I_{DD}$	operating modes	$V_{DD} = 5.5\text{ V}$ ; $f_{CLK} = 16\text{ MHz}$ notes 1 and 8			
	version P89CE528		—	50	mA
	version P80CE528		—	40	mA
	version P83CE528		—	40	mA
$I_{ID}$	idle mode	$V_{DD} = 5.5\text{ V}$ ; $f_{CLK} = 16\text{ MHz}$ notes 2 and 8			
	version P89CE528		—	10	mA
	version P80CE528		—	8	mA
	version P83CE528		—	8	mA
$I_{PD}$	power-down mode	$2\text{ V} \leq V_{PD} \leq V_{DD\text{ max}}$ note 3	—	100	µA
<b>Inputs</b>					
$V_{IL}$	LOW level input voltage (except EA, P1.6, P1.7)		−0.5	$0.2 V_{DD} - 0.1$	V
$V_{IL1}$	LOW level input voltage EA		−0.5	$0.2 V_{DD} - 0.3$	V
$V_{IL2}$	LOW level input voltage P1.6, P1.7	note 6	−0.5	$0.3 V_{DD}$	V
$V_{IH}$	HIGH level input voltage (except RST, XTAL1, P1.6, P1.7)		$0.2 V_{DD} + 0.9$	$V_{DD} + 0.5$	V

## 8-bit microcontroller with EMC and EEPROM P8xCE528

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
<b>Inputs</b>					
$V_{IH1}$	HIGH level input voltage RST, XTAL1		$0.7 V_{DD}$	$V_{DD} + 0.5$	V
$V_{IH2}$	HIGH level input voltage P1.6, P1.7	note 6	$0.7V_{DD}$	6.0	V
$I_{IL}$	input current logic 0 Ports 1, 2 and 3 (except P1.6 and P1.7)	$V_i = 0.45$ V	—	-50	$\mu$ A
$I_{TL}$	input current HIGH-to-LOW transition Ports 1, 2 and 3 (except P1.6 and P1.7)	$V_i = 2.0$ V	—	-650	$\mu$ A
$I_{L1}$	input leakage current Port 0, EA	$0.45 < V_i < V_{DD}$	—	$\pm 10$	$\mu$ A
$I_{L2}$	input leakage current P1.6 and P1.7	$0 V < V_i < 6 V$ $0 V < V_{DD} < 5.5 V$	—	$\pm 10$	$\mu$ A
<b>Outputs</b>					
$V_{OL}$	LOW level output voltage Ports 1, 2 and 3 (except P1.6 and P1.7)	$I_{OL} = 1.6$ mA; notes 4 and 7	—	0.45	V
$V_{OL1}$	LOW level output voltage Port 0, ALE, PSEN	$I_{OL} = 3.2$ mA; notes 4 and 7	—	0.45	V
$V_{OL2}$	LOW level output voltage P1.6 and P1.7	$I_{OL} = 3.0$ mA; note 7	—	0.40	V
$V_{OH}$	HIGH level output voltage Ports 1, 2, 3	$I_{OH} = -60 \mu$ A; $V_{DD} = 5 V \pm 10\%$ $I_{OH} = -25 \mu$ A $I_{OH} = -10 \mu$ A	2.4 $0.75V_{DD}$ $0.9V_{DD}$	— — —	V
$V_{OH1}$	HIGH level output voltage Port 0 in external Bus mode, ALE, PSEN, RST	$I_{OH} = -800 \mu$ A; $V_{DD} = 5 V \pm 10\%$ $I_{OH} = -300 \mu$ A; $I_{OH} = -80 \mu$ A; note 5	2.4 $0.75V_{DD}$ $0.9V_{DD}$	— — —	V
$R_{RST}$	RST pull-down resistor		50	150	k $\Omega$
$C_{VO}$	capacitance of input buffer	test frequency = 1 MHz; $T_{amb} = 25$ °C	—	10	pF

**Notes to DC Characteristics (EBx)**

- The operating supply current is measured with all output pins disconnected;  
XTAL1 driven with  $t_r = t_f = 5$  ns;  
 $V_{IL} = V_{SS} + 0.5$  V;  
 $V_{IH} = V_{DD} - 0.5$  V;  
XTAL2 not connected;  
EA = RST = Port 0 = P1.6 = P1.7 =  $V_{DD}$ ;  
the WDT is disabled (by the external RESET).

## 8-bit microcontroller with EMC and EEPROM P8xCE528

2. The Idle mode supply current is measured with all output pins disconnected;

XTAL1 driven with  $t_f = t_r = 5 \text{ ns}$ ;

$$V_{IL} = V_{SS} + 0.5 \text{ V};$$

$$V_{IH} = V_{DD} - 0.5 \text{ V};$$

XTAL2 not connected;

the WDT is disabled;

$$E\bar{A} = RST = V_{SS};$$

$$\text{Port } 0 = P1.6 = P1.7 = V_{DD}.$$

3. The Power-down current is measured with all output pins disconnected;

XTAL2 not connected;

WDT is disabled;

$$E\bar{A} = RST = XTAL1 = V_{SS};$$

$$\text{Port } 0 = P1.6 = P1.7 = V_{DD}.$$

4. Capacitive loading on Port 0 and Port 2 may cause spurious noise pulses to be superimposed on the LOW level output voltage of ALE, Port 1 and Port 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make a HIGH-to-LOW transition during bus operations. In the worst cases (capacitive loading  $> 100\text{pF}$ ), the noise pulse on the ALE line may exceed  $0.8 \text{ V}$ . In such cases it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.

5. Capacitive loading on Port 0 and Port 2 may cause the HIGH level output voltage on ALE and  $\overline{\text{PSEN}}$  to momentarily fall below the  $0.9 V_{DD}$  specification when the address bits are stabilizing.

6. The input threshold voltage of P1.6 and P1.7 (SIO1) meets the I<sup>2</sup>C-bus specification, so a voltage below  $0.3 V_{DD}$  will be recognized as a logic 0 while an input above  $0.7 V_{DD}$  will be recognized as a logic 1.

7. Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:

Maximum  $I_{OL}$  per port pin:  $10 \text{ mA}$ .

Maximum  $I_{OL}$  per 8-bit port: Port 0:  $26 \text{ mA}$ ; Ports 1, 2 and 3:  $15 \text{ mA}$ .

Maximum total  $I_{OL}$  for all output pins:  $71 \text{ mA}$ .

If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

8.  $I_{DD \text{ max.}}$  at other frequencies can be derived from Figure 15, where  $f$  is the external oscillator frequency in MHz;  $I_{DD \text{ max.}}$  is given in mA.

Symbol	Parameter	Conditions	Typical Value	Unit
$V_{OH}$	HIGH level output voltage	Port 0 in external bus mode, ALE, $\overline{\text{PSEN}}$ , RST note 8	$I_{OH} = -800 \mu\text{A}; V_{OH} = 2 \text{ V} \pm 10\%$ $I_{OH} = -300 \mu\text{A};$ $I_{OH} = -80 \mu\text{A};$ note 8	V
$V_{OL}$	LOW level output voltage	Port 0, ALE, $\overline{\text{PSEN}}$ note 7	$I_{OL} = -10 \mu\text{A}$ $I_{OL} = -25 \mu\text{A}$ note 8	V
$I_{DD}$	Supply current	XTAL1 driven with $f = 1 \text{ MHz}$ the WDT is disabled (by the external RESET)	$I_{DD} = 0.35 \text{ mA}$ $I_{DD} = 0.3 \text{ mA}$ $I_{DD} = 0.25 \text{ mA}$	mA
$R_{ST}$	RST pull-down resistor		50	k $\Omega$
$C_{IN}$	capacitance of input buffer	test frequency = $1 \text{ MHz}$ $T_{amb} = 25^\circ\text{C}$	10	pF

Notes to DC Characteristics (E8x)

1. The operating supply current is measured with all output pins disconnected.

2. XTAL1 driven with  $f = 1 \text{ MHz}$ .

$$V_{SS} = V_{DD} - 0.5 \text{ V};$$

$$V_{IL} = V_{SS} - 0.5 \text{ V};$$

$$V_{IH} = V_{DD} - 0.5 \text{ V};$$

XTAL2 not connected;

$$E\bar{A} = RST = \text{Port } 0 = P1.6 = P1.7 = V_{DD};$$

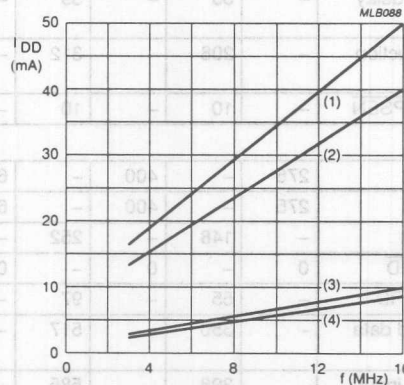
the WDT is disabled (by the external RESET).

## 8-bit microcontroller with EMC and FEEPROM P8xCE528

## 11 DC CHARACTERISTICS (EFx)

$V_{DD} = 5\text{ V} \pm 10\%$ ;  $V_{SS} = 0\text{ V}$ ;  $T_{amb} = -40$  to  $+85^\circ\text{C}$  (extended temperature range). All voltages with respect to  $V_{SS}$  unless otherwise specified. DC parameters not included here are the same as for the EBx temperature range data sheets.

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
<b>Inputs</b>					
$V_{IL}$	LOW level input voltage (except $\overline{EA}$ , P1.6, P1.7)		-0.5	0.2 $V_{DD}-0.15$	V
$V_{IL1}$	LOW level input voltage $\overline{EA}$		-0.5	0.2 $V_{DD}-0.35$	V
$V_{IH}$	HIGH level input voltage (except RST, XTAL1, P1.6, P1.7)		0.2 $V_{DD}$ +1.0	$V_{DD}+0.5$	V
$V_{IH1}$	HIGH level input voltage RST, XTAL1		0.7 $V_{DD}$ +0.1	$V_{DD}+0.5$	V
$I_{IL}$	input current logic 0 Ports 1, 2 and 3 (except P1.6 and P1.7)	$V_I = 0.45\text{ V}$	-	-75	$\mu\text{A}$
$I_{TL}$	input current HIGH-to-LOW transition Ports 1, 2 and 3 (except P1.6 and P1.7)	$V_I = 2.0\text{ V}$	-	-750	$\mu\text{A}$



$V_{DD} = 5.5\text{ V}$

Valid only within frequency specifications of device under test.

- (1) Maximum operating mode P89CE528.
- (2) Maximum operating mode P83CE528 and P80CE528.
- (3) Maximum idle mode P89CE528.
- (4) Maximum idle mode P83CE528 and P80CE528.

Fig.15 Supply current  $I_{DD}$  as a function of frequency.

## 8-bit microcontroller with EMC and FEEPROM P8xCE528

## 12 AC CHARACTERISTICS

EBX:  $V_{DD} = 5\text{ V} \pm 10\%$ ;  $V_{SS} = 0\text{ V}$ ;  $T_{amb} = 0\text{ to }+70\text{ }^{\circ}\text{C}$ ;  $t_{CLCL}\text{ min.} = 63\text{ ns}$ .

EFX:  $V_{DD} = 5\text{ V} \pm 10\%$ ;  $V_{SS} = 0\text{ V}$ ;  $T_{amb} = -40\text{ to }+85\text{ }^{\circ}\text{C}$ ;  $t_{CLCL}\text{ min.} = 63\text{ ns}$ .

All versions EXX:  $CI = 100\text{ pF}$  for Port 0, ALE and PSEN;  $CI = 80\text{ pF}$  for all other outputs unless otherwise specified.

$t_{CLCL}\text{ min.} = 1/f\text{ max.}$  (maximum operating frequency).

SYMBOL	PARAMETER	16 MHz		12 MHz		VARIABLE CLOCK		UNIT
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
$1/t_{CLCL}$	oscillator frequency	—	—	—	—	3.5	16	MHz
$t_{LHLL}$	ALE pulse duration	85	—	127	—	$2 t_{CLCL}-40$	—	ns
$t_{AVLL}$	address set-up time to ALE	8	—	28	—	$t_{CLCL}-55$	—	ns
$t_{LLAX}$	address hold time after ALE	28	—	48	—	$t_{CLCL}-35$	—	ns
$t_{LLIV}$	time from ALE to valid instruction input	—	150	—	233	—	$4 t_{CLCL}-100$	ns
$t_{LLPL}$	time from ALE to control pulse PSEN	23	—	43	—	$t_{CLCL}-40$	—	ns
$t_{PLPH}$	control pulse duration PSEN	143	—	205	—	$3 t_{CLCL}-45$	—	ns
$t_{PLIV}$	time from PSEN to valid instruction input	—	83	—	145	—	$3 t_{CLCL}-105$	ns
$t_{PXIX}$	input instruction hold time after PSEN	0	—	0	—	0	—	ns
$t_{PXIZ}$	input instruction float delay after PSEN	—	38	—	59	—	$t_{CLCL}-25$	ns
$t_{AVIV}$	address to valid instruction input	—	208	—	312	—	$5 t_{CLCL}-105$	ns
$t_{PLAZ}$	address float time to PSEN	—	10	—	10	—	10	ns
<b>Data memory</b>								
$t_{RLRH}$	$\overline{RD}$ pulse duration	275	—	400	—	$6 t_{CLCL}-100$	—	ns
$t_{WLWH}$	$\overline{WR}$ pulse duration	275	—	400	—	$6 t_{CLCL}-100$	—	ns
$t_{RLDV}$	$\overline{RD}$ to valid data input	—	148	—	252	—	$5 t_{CLCL}-165$	ns
$t_{RHDX}$	data hold time after $\overline{RD}$	0	—	0	—	0	—	ns
$t_{RHDX}$	data float delay after $\overline{RD}$	—	55	—	97	—	$2 t_{CLCL}-70$	ns
$t_{LLDV}$	time from ALE to valid data input	—	350	—	517	—	$8 t_{CLCL}-150$	ns
$t_{AVDV}$	address to valid data input	—	398	—	585	—	$9 t_{CLCL}-165$	ns
$t_{LLWL}$	time from ALE to $\overline{RD}$ or $\overline{WR}$	138	238	200	300	$3 t_{CLCL}-50$	$3 t_{CLCL}+50$	ns
$t_{AVWL}$	time from address to $\overline{RD}$ or $\overline{WR}$	120	—	203	—	$4 t_{CLCL}-130$	—	ns
$t_{QVWX}$	data valid to $\overline{WR}$ transition	3	—	23	—	$t_{CLCL}-60$	—	ns
$t_{WHQX}$	data hold time after $\overline{WR}$	13	—	33	—	$t_{CLCL}-50$	—	ns
$t_{RLAZ}$	address float delay after $\overline{RD}$	—	0	—	0	—	0	ns
$t_{WHLH}$	time from $\overline{RD}$ or $\overline{WR}$ HIGH to ALE HIGH	23	103	43	123	$t_{CLCL}-40$	$t_{CLCL}+40$	ns



# 8-bit microcontroller with EMC and FEEPROM s P8xCE528

SYMBOL	PARAMETER	16 MHz		12 MHz		VARIABLE CLOCK		UNIT
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
External clock								
t <sub>CHCK</sub>	HIGH time	20	—	—	—	20	—	ns
t <sub>CLCX</sub>	LOW time	20	—	—	—	20	—	ns
t <sub>CLCH</sub>	rise time	—	20	—	—	—	20	ns
t <sub>CHCL</sub>	fall time	—	20	—	—	—	20	ns
UART timing shift register mode								
t <sub>XLXL</sub>	serial port clock cycle time	750	—	1000	—	12t <sub>CLCL</sub>	—	ns
t <sub>QVXH</sub>	output data setup to clock rising edge	492	—	700	—	10t <sub>CLCL</sub> - 133	—	ns
t <sub>XHQX</sub>	output data hold after clock rising edge	8	—	50	—	2t <sub>CLCL</sub> - 117	—	ns
t <sub>XHDX</sub>	input data hold after clock rising edge	0	—	0	—	0	—	ns
t <sub>XHDV</sub>	clock rising edge to input data valid	—	492	—	700	—	10t <sub>CLCL</sub> - 133	ns

- Notes to PC-bus Characteristics (bit-level)
- At  $f_{CLK} = 3.5$  MHz, this evaluates to  $14 \times 286$  ns = 4  $\mu$ s, i.e. the bit-level PC-bus interface can respond to the PC-bus protocol for  $f_{CLK} \leq 3.5$  MHz.
  - This parameter is determined by the user software. It has to comply with the PC-bus specification.
  - This value gives the auto-clock pulse length which meets the PC-bus specification for the specified XTAL1 clock frequency range. Alternatively, the SCL pulse may be timed by software.
  - Spikes on SDA and SCL lines with a duration of less than  $4 \times t_{CLCL}$  will be filtered out.
  - The rise time is determined by the external bus line capacitance and pull-up resistor. It must be  $\leq 1$   $\mu$ s.
  - The maximum capacitance on bus lines SDA and SCL is 400 pF.

## 8-bit microcontroller with EMC and EEPROM P8xCE528

13 I<sup>2</sup>C-bus CHARACTERISTICS (bit-level)

$t_{\text{CLCL}} = 1/f_{\text{CLK}} = \text{one oscillator period at pin XTAL1}$ . For  $63 \text{ ns} < t_{\text{CLCL}} < 286 \text{ ns}$  ( $16 \text{ MHz} > f_{\text{OSC}} > 3.5 \text{ MHz}$ ) the I<sup>2</sup>C interface meets the I<sup>2</sup>C-bus specification for bit rates up to 100 kbit/s.

SYMBOL	PARAMETER	INPUT	OUTPUT	I <sup>2</sup> C-bus Spec.	UNIT
<b>SCL timing</b>					
$t_{\text{HD:STA}}$	START condition hold time	$\geq 14 t_{\text{CLCL}}$ ; note 1	note 2	$\geq 4.0$	$\mu\text{s}$
$t_{\text{LOW}}$	SCL LOW time	$\geq 16 t_{\text{CLCL}}$	note 2	$\geq 4.7$	$\mu\text{s}$
$t_{\text{HIGH}}$	SCL HIGH time	$\geq 14 t_{\text{CLCL}}$ ; note 1	$\geq 80 t_{\text{CLCL}}$ ; note 3	$\geq 4.0$	$\mu\text{s}$
$t_{\text{RC}}$	SCL rise time	$\leq 1 \mu\text{s}$ ; note 4	note 5	$\leq 1.0$	$\mu\text{s}$
$t_{\text{FC}}$	SCL fall time	$\leq 0.3 \mu\text{s}$ ; note 4	$\leq 0.3 \mu\text{s}$ ; note 6	$\leq 0.3$	$\mu\text{s}$
<b>SDA timing</b>					
$t_{\text{SU:DAT}}$	data set-up time	$\geq 250 \text{ ns}$	note 2	$\geq 250$	ns
$t_{\text{HD:DAT}}$	data hold time	$\geq 0 \text{ ns}$	note 2	$\geq 0$	ns
$t_{\text{SU:STA}}$	repeated START set-up time	$\geq 14 t_{\text{CLCL}}$ ; note 1	note 2	$\geq 4.7$	$\mu\text{s}$
$t_{\text{SU:STO}}$	STOP condition set-up time	$\geq 14 t_{\text{CLCL}}$ ; note 1	note 2	$\geq 4.0$	$\mu\text{s}$
$t_{\text{BUF}}$	bus free time	$\geq 14 t_{\text{CLCL}}$ ; note 1	note 2	$\geq 4.7$	$\mu\text{s}$
$t_{\text{RD}}$	SDA rise time	$\leq 1 \mu\text{s}$ ; note 4	note 5	$\leq 1.0$	$\mu\text{s}$
$t_{\text{FD}}$	SDA fall time	$\leq 0.3 \mu\text{s}$ ; note 4	$\leq 0.3 \mu\text{s}$ ; note 6	$\leq 0.3$	$\mu\text{s}$

Notes to I<sup>2</sup>C-bus Characteristics (bit-level)

- At  $f_{\text{CLK}} = 3.5 \text{ MHz}$ , this evaluates to  $14 \times 286 \text{ ns} = 4 \mu\text{s}$ , i.e. the bit-level I<sup>2</sup>C-bus interface can respond to the I<sup>2</sup>C-bus protocol for  $f_{\text{CLK}} \geq 3.5 \text{ MHz}$ .
- This parameter is determined by the user software, it has to comply with the I<sup>2</sup>C-bus specification.
- This value gives the auto-clock pulse length which meets the I<sup>2</sup>C-bus specification for the specified XTAL1 clock frequency range. Alternatively, the SCL pulse may be timed by software.
- Spikes on SDA and SCL lines with a duration of less than  $4 \times t_{\text{CLCL}}$  will be filtered out.
- The rise time is determined by the external bus line capacitance and pull-up resistor, it must be  $\leq 1 \mu\text{s}$ .
- The maximum capacitance on bus lines SDA and SCL is 400 pF.

## 8-bit microcontroller with EMC and FEEPROM P8xCE528

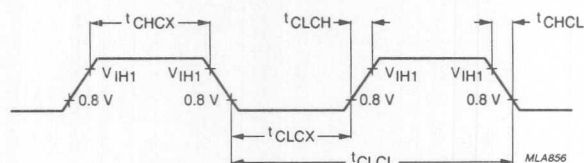
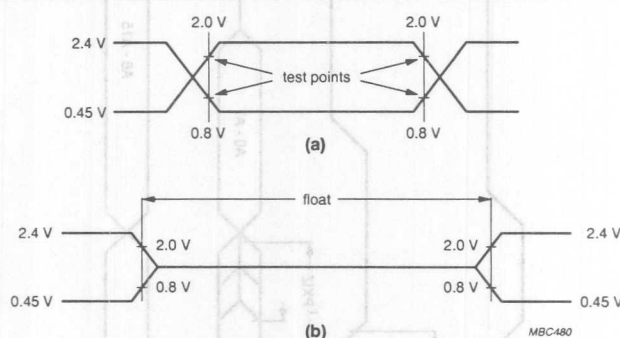


Fig.16 External clock drive XTAL1.



AC testing inputs are driven at 2.4 V for a logic 1 and 0.45 V for a logic 0. Timing measurements are taken at 2.0 V for a logic 1 and 0.8 V for logic 0 see (a). The float state is defined as the point at which a Port 0 pin sinks 3.2 mA or sources 400  $\mu$ A at the voltage test levels see (b).

Fig.17 AC testing input, output waveform (a) and float waveform (b).

## 14.1 Timing symbol definitions

Oscillator:

$f_{CLK}$  = clock frequency

$t_{CLCL}$  = clock period

Timing symbols (acronyms):

Each timing symbol has five characters. The first character is always a "t" (= time). the remaining four characters of the symbol (typed in subscript), depending on their relative positions, indicate the name of a signal or the logical status of that signal. the designations are as follows:

A = address

C = clock

D = input data

H = logic level HIGH

I = instruction (program memory contents)

L = Logic level LOW or ALE

P =  $\overline{PSEN}$

Q = output data

R =  $\overline{RD}$  signal

t = time

V = valid

W =  $\overline{WR}$  signal

X = no longer a valid logic level

Z = float

Examples:

$t_{AVLL}$  = time for address valid to ALE LOW

$t_{LLPL}$  = time for ALE LOW to  $\overline{PSEN}$  LOW

## 8-bit microcontroller with EMC and FEEPROMs P8xCE528

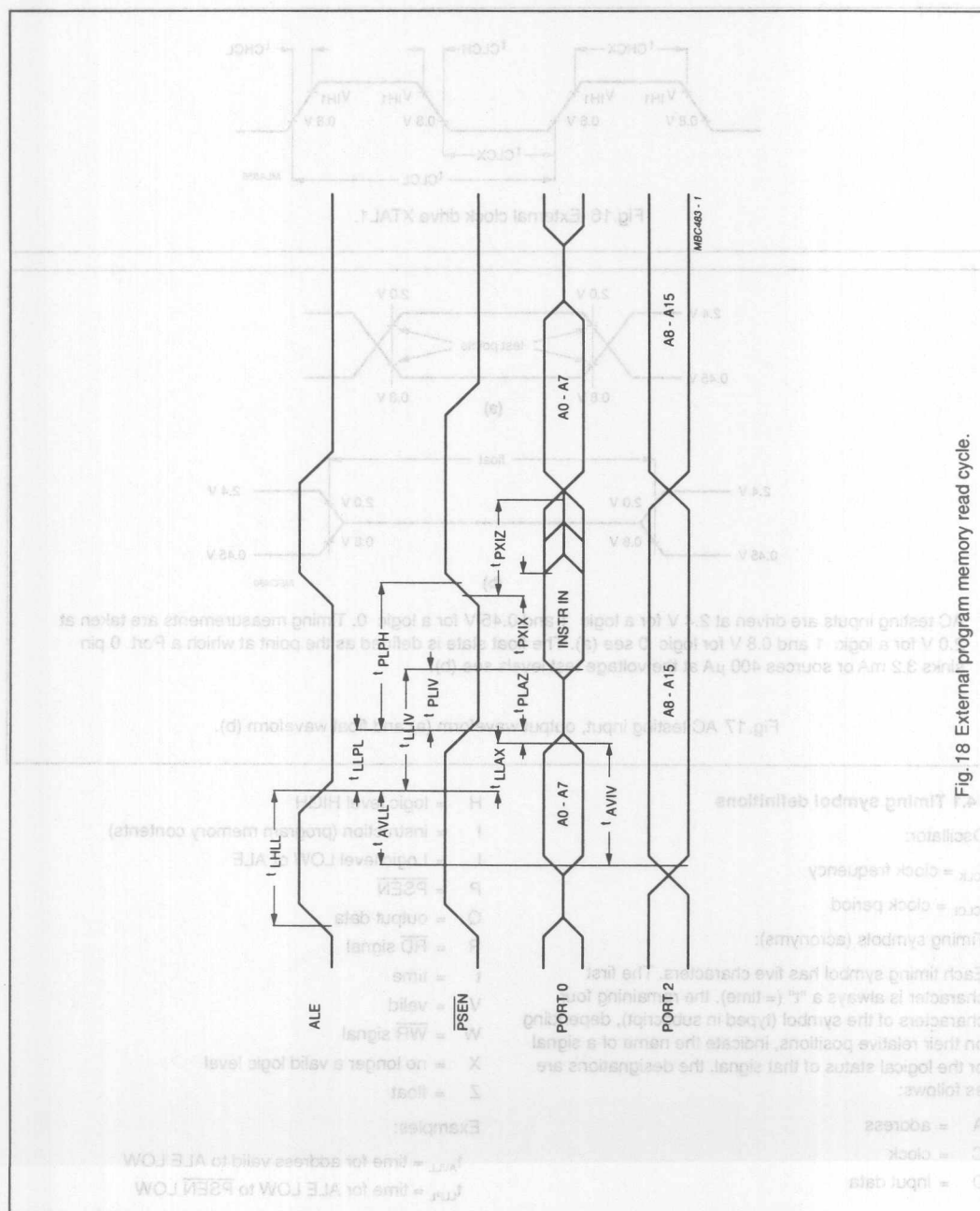


Fig. 18 External program memory read cycle.

## 8-bit microcontroller with EMC and FEEPROM P8xCE528

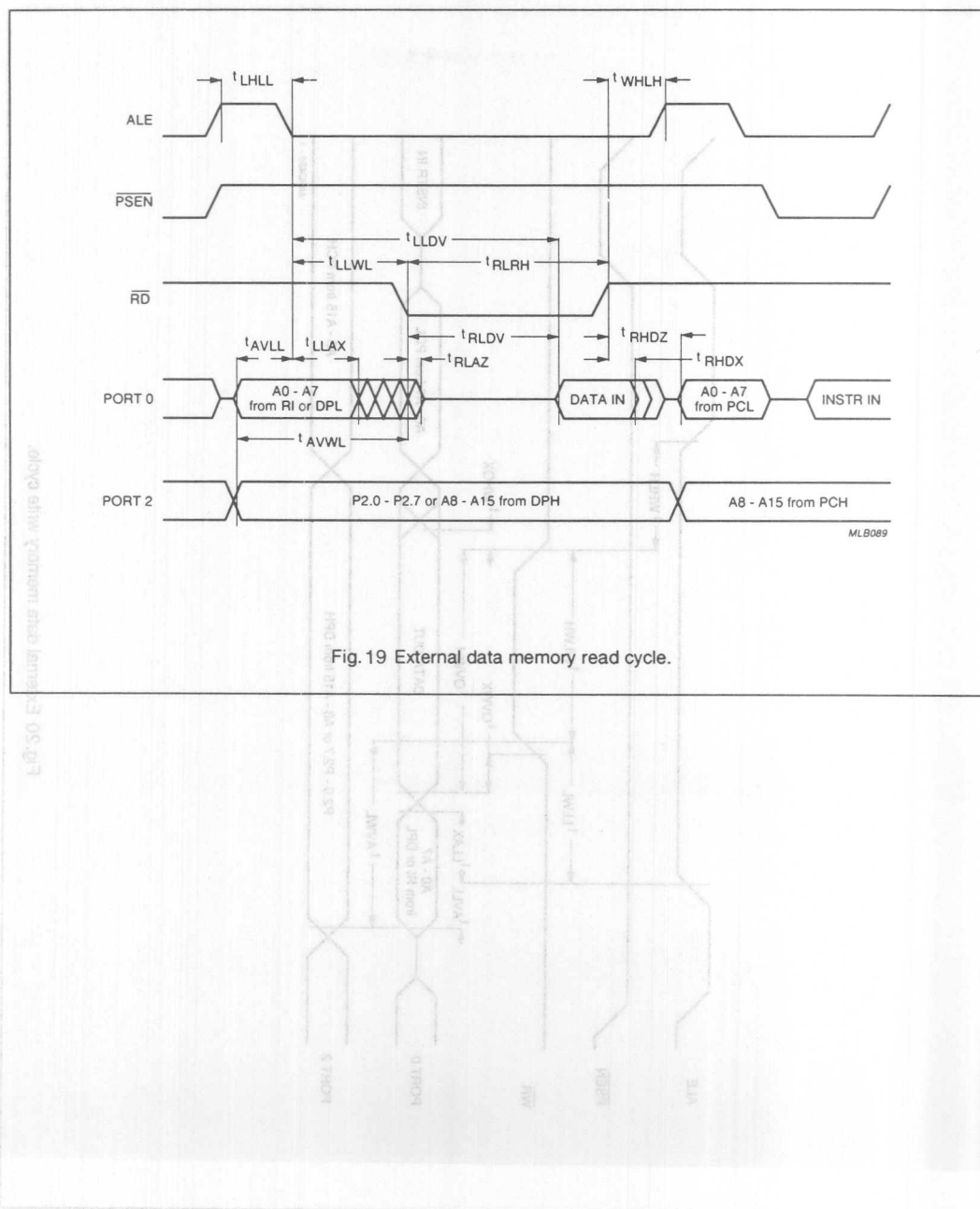


Fig.19 External data memory read cycle.



## 8-bit microcontroller with EMC and EEPROM P8xCE528

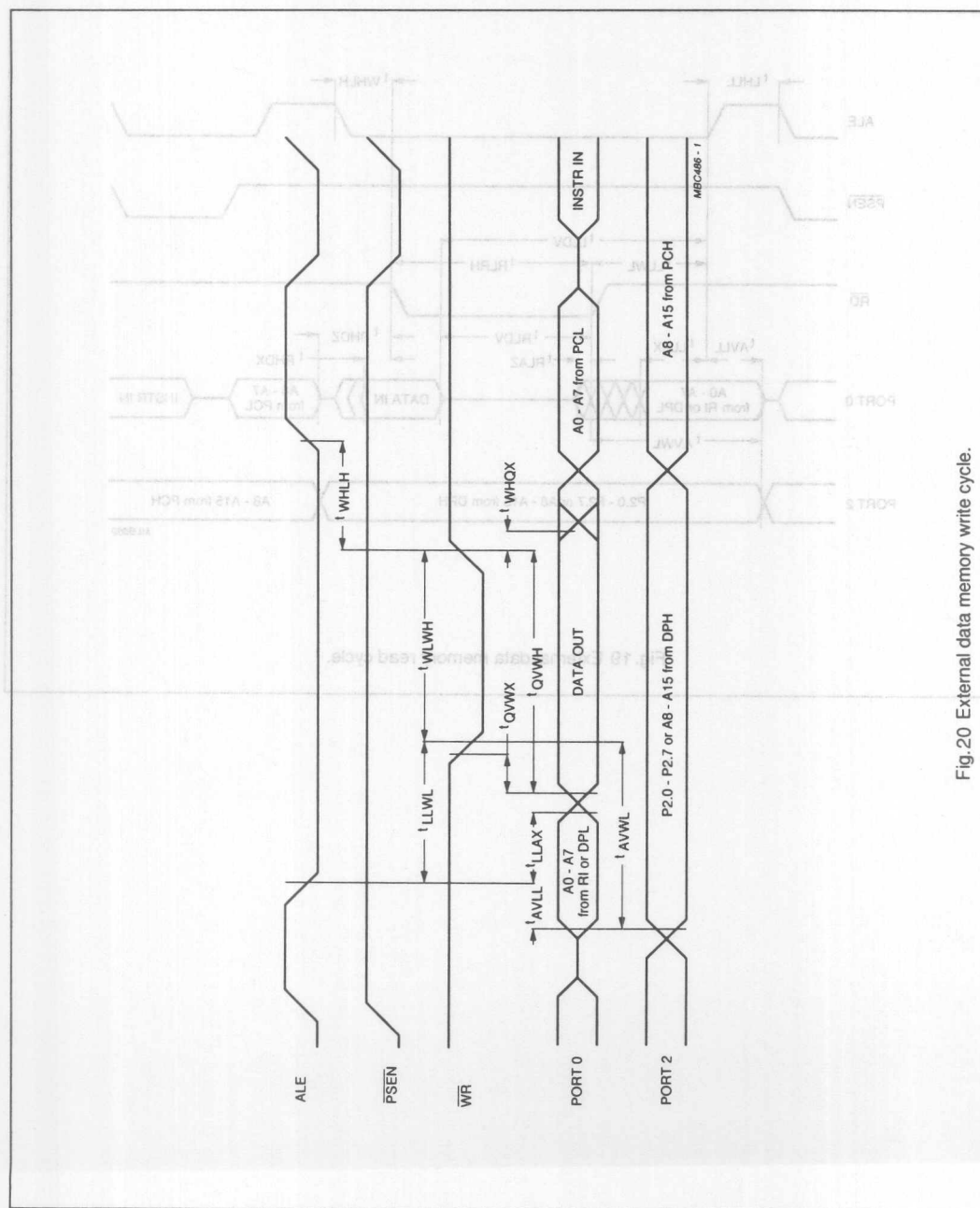


Fig.20 External data memory write cycle.

# 8-bit microcontroller with EMC and EEPROM P8xC5E28

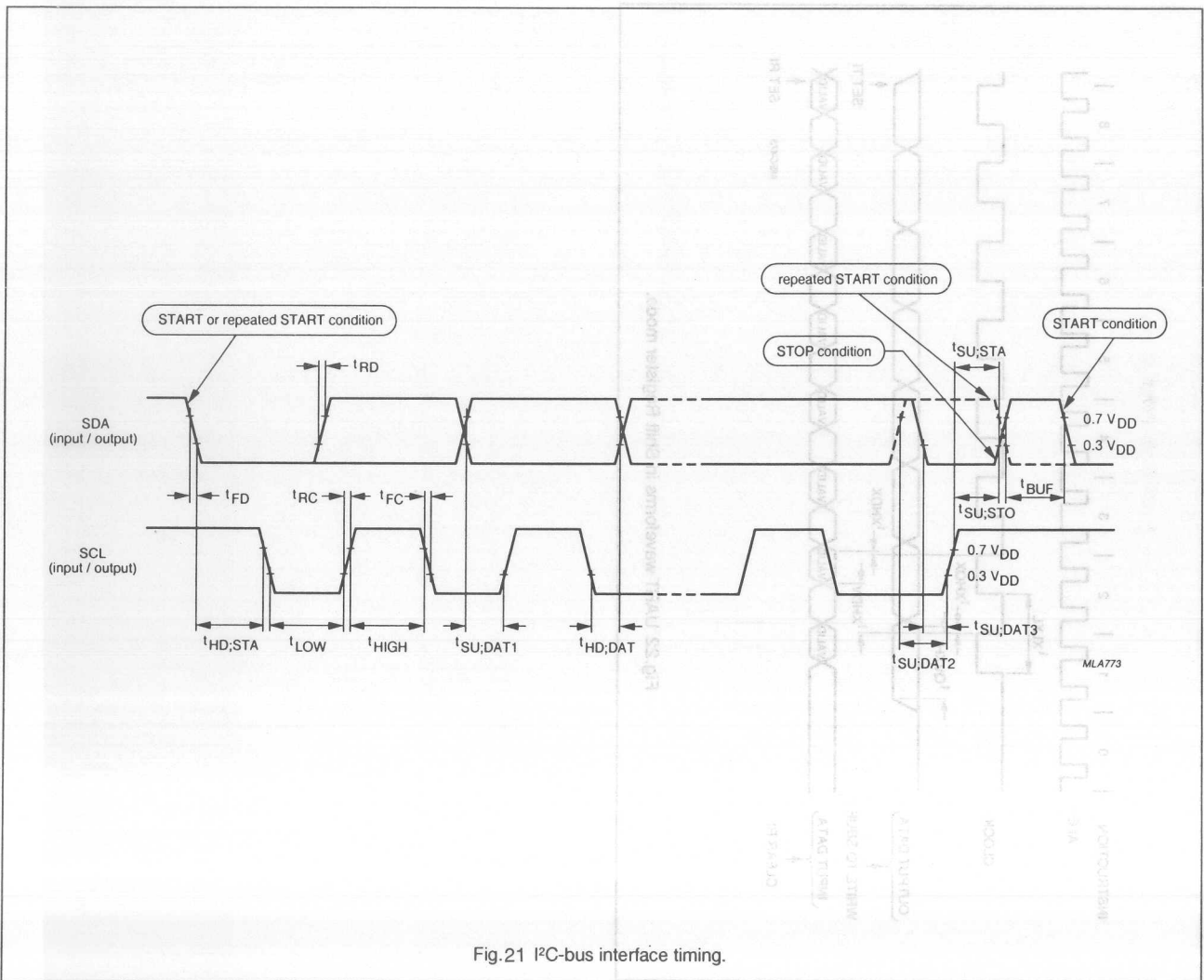


Fig.21 I²C-bus interface timing.

## 8-bit microcontroller with EMC and FEEPROM P8xCE528

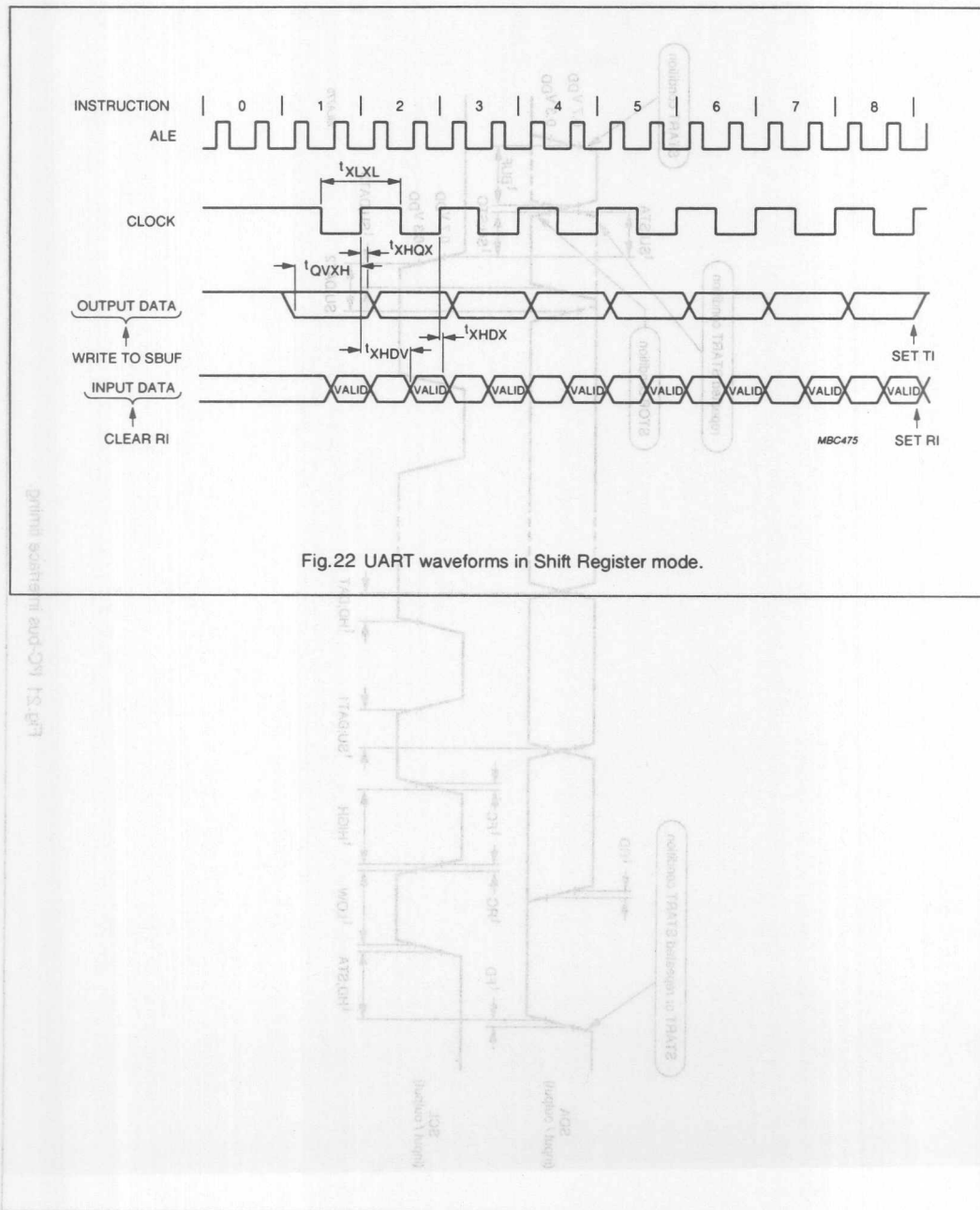


Fig.22 UART waveforms in Shift Register mode.

## 8-bit microcontroller with EMC and FEEPROM

P8xCE528

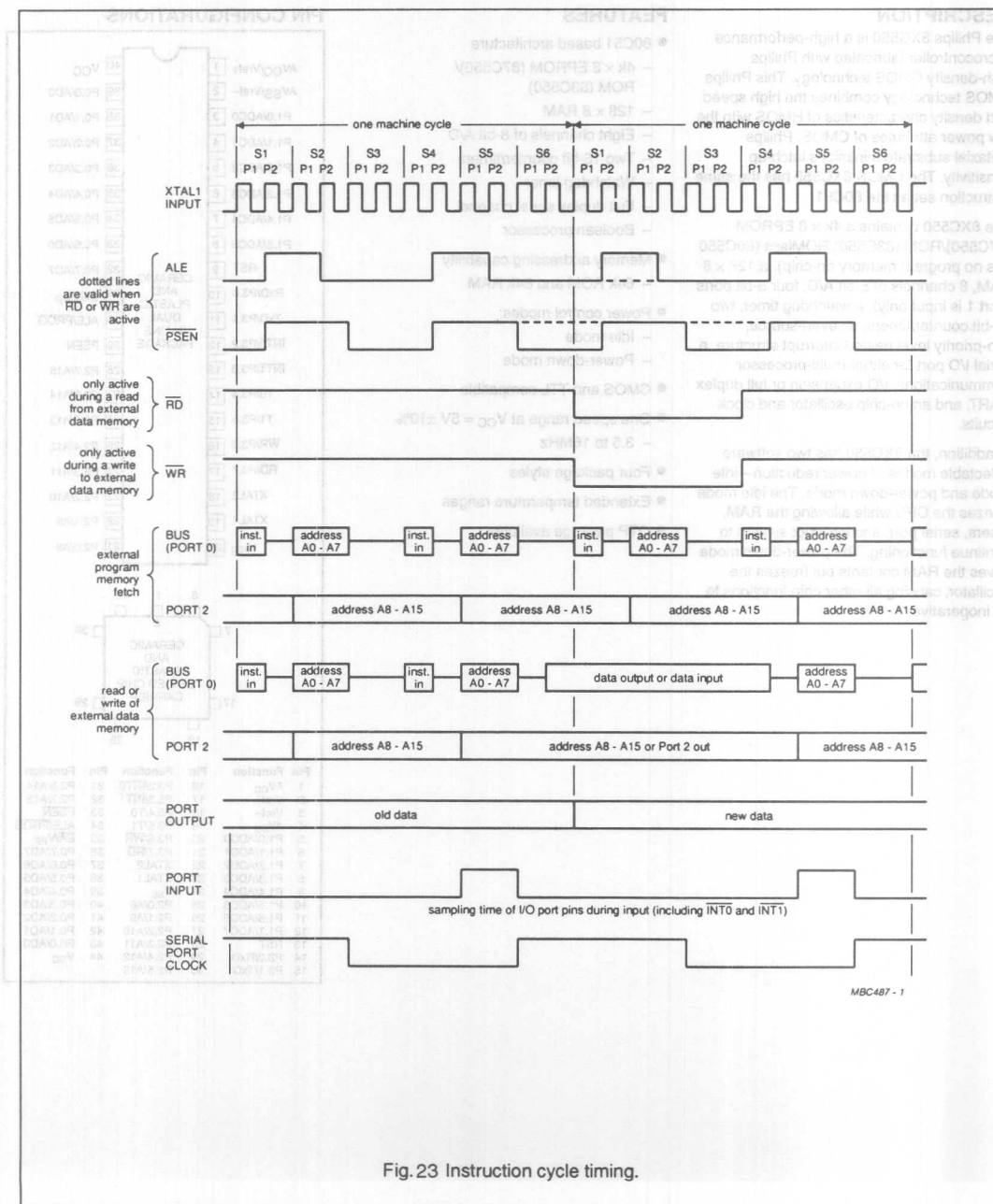


Fig.23 Instruction cycle timing.

# CMOS single-chip 8-bit microcontroller with A/D and watchdog timer

80C550/83C550/87C550

## DESCRIPTION

The Philips 8XC550 is a high-performance microcontroller fabricated with Philips high-density CMOS technology. This Philips CMOS technology combines the high speed and density characteristics of HMOS with the low power attributes of CMOS. Philips epitaxial substrate minimizes latch-up sensitivity. The CMOS 8XC550 has the same instruction set as the 80C51.

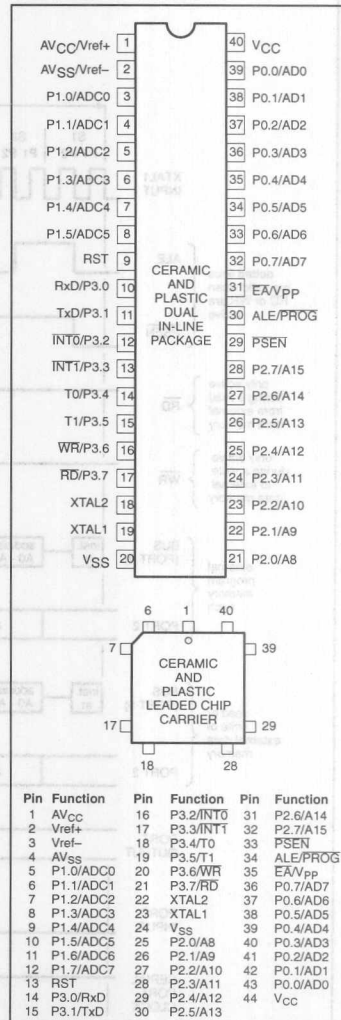
The 8XC550 contains a 4k × 8 EPROM (87C550)/ROM (83C550)/ROMless (80C550) has no program memory on-chip), a 128 × 8 RAM, 8 channels of 8-bit A/D, four 8-bit ports (port 1 is input only), a watchdog timer, two 16-bit counter/timers, a seven-source, two-priority level nested interrupt structure, a serial I/O port for either multi-processor communications, I/O expansion or full duplex UART, and an on-chip oscillator and clock circuits.

In addition, the 8XC550 has two software selectable modes of power reduction—idle mode and power-down mode. The idle mode freezes the CPU while allowing the RAM, timers, serial port, and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.

## FEATURES

- 80C51 based architecture
  - 4k × 8 EPROM (87C550)/ROM (83C550)
  - 128 × 8 RAM
  - Eight channels of 8-bit A/D
  - Two 16-bit counter/timers
  - Watchdog timer
  - Full duplex serial channel
  - Boolean processor
- Memory addressing capability
  - 64k ROM and 64k RAM
- Power control modes:
  - Idle mode
  - Power-down mode
- CMOS and TTL compatible
- One speed range at  $V_{CC} = 5V \pm 10\%$ 
  - 3.5 to 16MHz
- Four package styles
- Extended temperature ranges
- OTP package available

## PIN CONFIGURATIONS





# CMOS single-chip 8-bit microcontroller with A/D and watchdog timer

## 80C550/83C550/87C550

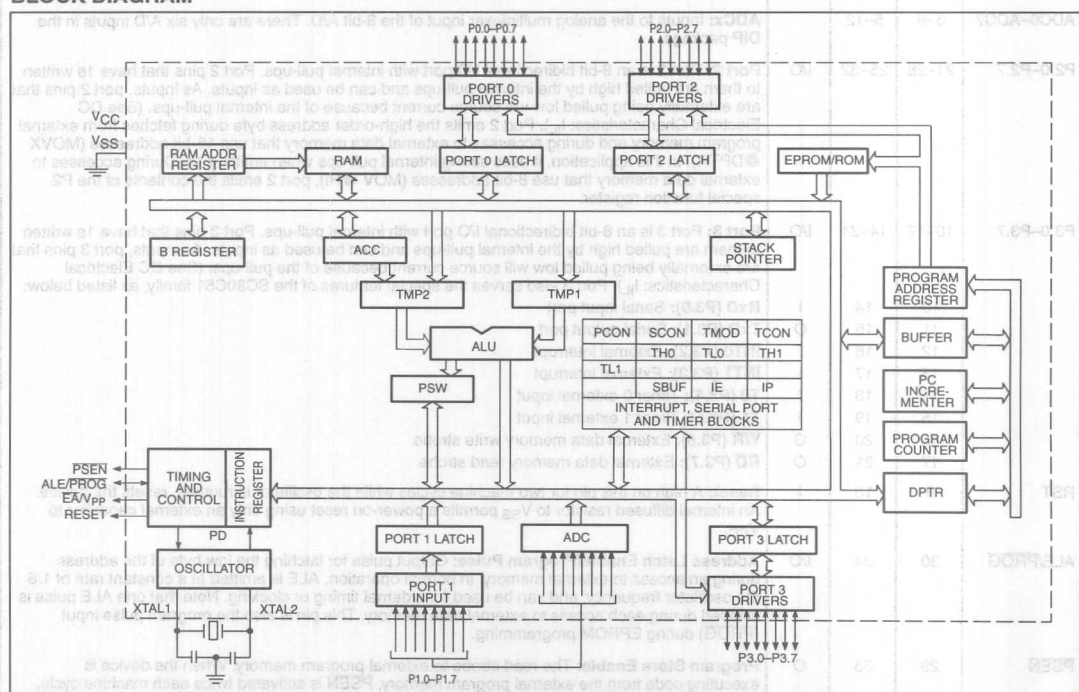
### ORDERING INFORMATION

ROMless	ROM	EPROM	TEMPERATURE RANGE °C AND PACKAGE <sup>1</sup>	FREQ MHz	DRAWING NUMBER
		P87C550EBF FA UV	0 to +70, Ceramic Dual In-Line Package	3.5 to 16	0590B
		P87C550EBL KA UV	0 to +70, Ceramic Leaded Chip Carrier	3.5 to 16	1472A
P80C550EBP N	P83C550EBP N	P87C550EBP N OTP	0 to +70, Plastic Dual In-Line Package	3.5 to 16	0415C
P80C550EBA A	P83C550EBA A	P87C550EBA A OTP	0 to +70, Plastic Leaded Chip Carrier	3.5 to 16	0403G
P80C550EFP N	P83C550EFP N	P87C550EFP N OTP	-40 to +85, Plastic Dual In-Line Package	3.5 to 16	0415C
P80C550EFA A	P83C550EFA A	P87C550EFA A OTP	-40 to +85, Plastic Leaded Chip Carrier	3.5 to 16	0403G
		P87C550EFL KA UV	-40 to +85, Ceramic Leaded Chip Carrier	3.5 to 16	1472A
		P87C550EFF FA UV	-40 to +85, Ceramic Dual In-Line Package	3.5 to 16	0590B

### NOTES:

1. OTP = One Time Programmable EPROM. UV = UV Erasable EPROM.

### BLOCK DIAGRAM



# CMOS single-chip 8-bit microcontroller with A/D and watchdog timer

80C550/83C550/87C550

## PIN DESCRIPTION

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	LCC		
V <sub>SS</sub>	20	24	I	<b>Ground:</b> 0V reference.
V <sub>CC</sub>	40	44	I	<b>Power Supply:</b> This is the power supply voltage for normal, idle, and power-down operation.
AV <sub>CC</sub>	1	1	I	<b>Analog Power Supply:</b> Analog supply voltage.
AV <sub>SS</sub>	2	4	I	<b>Analog Ground:</b> Analog 0V reference.
V <sub>ref+</sub>		2	I	<b>V<sub>ref</sub>:</b> A/D converter reference level inputs. Note that these references are combined with AV <sub>CC</sub> and AV <sub>SS</sub> in the 40-pin DIP package.
V <sub>ref-</sub>		3	I	
P0.0–0.7	39–32	43–36	I/O	<b>Port 0:</b> Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s. Port 0 also outputs the code bytes during program verification in the S87C550. External pull-ups are required during program verification.
P1.0–P1.7	3–8	5–12	I	<b>Port 1:</b> Port 1 is an 8-bit input only port (6-bit in the DIP package; bits P1.6 and P1.7 are not implemented). Port 1 digital input can be read out any time.
ADC0–ADC7	3–8	5–12		<b>ADCx:</b> Inputs to the analog multiplexer input of the 8-bit A/D. There are only six A/D inputs in the DIP package.
P2.0–P2.7	21–28	25–32	I/O	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register.
P3.0–P3.7	10–17	14–21	I/O	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 3 also serves the special features of the SC80C51 family, as listed below:
	10	14	I	<b>RxD (P3.0):</b> Serial input port
	11	15	O	<b>TxD (P3.1):</b> Serial output port
	12	16	I	<b>INT0 (P3.2):</b> External interrupt
	13	17	I	<b>INT1 (P3.3):</b> External interrupt
	14	18	I	<b>T0 (P3.4):</b> Timer 0 external input
	15	19	I	<b>T1 (P3.5):</b> Timer 1 external input
	16	20	O	<b>WR (P3.6):</b> External data memory write strobe
	17	21	O	<b>RD (P3.7):</b> External data memory read strobe
RST	9	13	I	<b>Reset:</b> A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to V <sub>SS</sub> permits a power-on reset using only an external capacitor to V <sub>CC</sub> .
ALE/PROG	30	34	I/O	<b>Address Latch Enable/Program Pulse:</b> Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory. This pin is also the program pulse input (PROG) during EPROM programming.
PSEN	29	33	O	<b>Program Store Enable:</b> The read strobe to external program memory. When the device is executing code from the external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory. PSEN is not activated during fetches from internal program memory.
EA/V <sub>PP</sub>	31	35	I	<b>External Access Enable/Programming Supply Voltage:</b> EA must be externally held low to enable the device to fetch code from external program memory locations 0000H to 0FFFH. If EA is held high, the device executes from internal program memory unless the program counter contains an address greater than 0FFFH. For the 80C550 ROMless part, EA must be held low for the part to operate properly. This pin also receives the 12.75V programming supply voltage (V <sub>PP</sub> ) during EPROM programming.
XTAL1	19	23	I	<b>Crystal 1:</b> Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	22	O	<b>Crystal 2:</b> Output from the inverting oscillator amplifier.

# CMOS single-chip 8-bit microcontroller with A/D and watchdog timer

80C550/83C550/87C550

Table 1. 8XC550 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION										RESET VALUE
			MSB								LSB		
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0		00H	
ADAT#	A/D result	C6H										xxH	
ADCON#	A/D control	C5H	—	—	—	ADCI	ADCS	AADR2	AADR1	AADR0		xxx00000B	
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0		00H	
DPTR:	Data pointer												
DPH	(2 bytes): High byte	83H										00H	
DPL	Low byte	82H										00H	
			BF	BE	BD	BC	BB	BA	B9	B8			
IP*#	Interrupt priority	B8H	—	PWD	PAD	PS	PT1	PX1	PT0	PX0		x0000000B	
			AF	AE	AD	AC	AB	AA	A9	A8			
IE*#	Interrupt enable	A8H	EA	EWD	EAD	ES	ET1	EX1	ET0	EX0		00H	
P0*	Port 0	80H	87	86	85	84	83	82	81	80		FFH	
P1*	Port 1	90H	97	96	95	94	93	92	91	90		FFH	
P2*	Port 2	A0H	A7	A6	A5	A4	A3	A2	A1	A0		FFH	
P3*	Port 3	B0H	B7	B6	B5	B4	B3	B2	B1	B0		FFH	
PCON#	Power control	87H	SMOD	SIDL	—	—	GF1	GF0	PD	IDL		00xx0000B	
			D7	D6	D5	D4	D3	D2	D1	D0			
PSW*	Program status word	D0H	CY	AC	F0	RS1	RS0	OV	—	P		00H	
SBUF	Serial data buffer	99H	9F	9E	9D	9C	9B	9A	99	98		xxH	
SCON*	Serial port control	98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI		00H	
SP	Stack pointer	81H	8F	8E	8D	8C	8B	8A	89	88		07H	
			8F	8E	8D	8C	8B	8A	89	88		00H	
TCON*	Timer counter/control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0		00H	
			GATE	C/T	M1	M0	GATE	C/T	M1	M0		00H	
TMOD	Timer/counter mode	89H										00H	
TH0	Timer 0 high byte	8CH										00H	
TH1	Timer 1 high byte	8DH										00H	
TL0	Timer 0 low byte	8AH										00H	
TL1	Timer 1 low byte	8BH										00H	
			C7	C6	C5	C4	C3	C2	C1	C0			
WDCON*#	Watchdog timer control	C0H	PRE2	PRE1	PRE0	—	—	WDRUN	WDT0F	WDMOD		000xx000B**	
WDL#	Watchdog timer reload	C1H										FFH**	
WFEED1#	Watchdog timer feed 1	C2H										xxH	
WFEED2#	Watchdog timer feed 2	C3H										xxH	

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs. A/D ADCS in ADCON# is bit addressable.

\*\*This value is not valid for a masked ROM part (83C550) when running from internal memory (EA = 1). See data sheet for details.

# CMOS single-chip 8-bit microcontroller with A/D and watchdog timer

80C550/83C550/87C550

MSB								LSB							
SMOD	SIDL	X	X	GF1	GF0	PD	IDL								
BIT	SYMBOL	FUNCTION													
PCON.7	SMOD	DOUBLE BAUD RATE													
PCON.7	SIDL	SERIAL PORT IDLE													
PCON.5	X	RESERVED FOR FUTURE USE													
PCON.4	X	RESERVED FOR FUTURE USE													
PCON.3	GF1	GENERAL PURPOSE FLAG BIT													
PCON.2	GF0	GENERAL PURPOSE FLAG BIT													
PCON.1	PD	POWER DOWN BIT													
PCON.0	IDL	IDLE MODE BIT													

NOTE:  
The PCON register is at SFR byte address 87H. Its contents following a reset are 00XX0000.

Figure 1. Power Control Register (PCON)

MSB					LSB				
X	X	X	ADCI	ADCS	AADR2	AADR1	AADR0		

INPUT CHANNEL SELECTION			
ADDR2	ADDR1	ADDR0	INPUT PIN
0	0	0	ADC0
0	0	1	ADC1
0	1	0	ADC2
0	1	1	ADC3
1	0	0	ADC4
1	0	1	ADC5
1	1	0	ADC6
1	1	1	ADC7

BIT	SYMBOL	FUNCTION
ADCON.7	—	NOT USED
ADCON.6	—	NOT USED
ADCON.5	—	NOT USED
ADCON.4	ADCI	ADC INTERRUPT FLAG. This flag is set when an ADC conversion result is ready to be read. An interrupt is invoked if the A/D interrupt is enabled. The flag must be cleared by software. It cannot be set by software.
ADCON.3	ADCS	ADC START AND STATUS. Setting this flag starts an A/D conversion. The ADC logic insures that this signal is high while the ADC is busy. On completion of the conversion, ADCS is reset at the same time the interrupt flag ADCI is set. ADCS cannot be reset by software.
ADCON.2	ADDR2	ANALOG INPUT SELECT 2
ADCON.1	ADDR1	ANALOG INPUT SELECT 1
ADCON.0	ADDR0	ANALOG INPUT SELECT 0

**Figure 2. A/D Control Register (ADCON)**

Figure 2. A/D Control Register (ADCON)

## A/D CONVERTER

The analog input circuitry consists of an 8-input analog multiplexer and an analog-to-digital converter with 8-bit resolution. In the LCC package, the analog reference voltage and analog power supplies are connected via separate input pins; in the DIP package, Vref+ is combined with AVCC and Vref- is combined with AVSS. The analog inputs are alternate functions to port 1, which is an input only port. Digital input to port 1 can be read any time during an A/D conversion. Care should be exercised in mixing analog and digital signals on port 1, because cross talk from the digital input signals can degrade the A/D conversion accuracy of the analog input. An A/D conversion requires 40 machine cycles.

The A/D converter is controlled by the ADCON special function register. The input channel to be converted is selected by the analog multiplexer by setting ADCON register bits, ADDR2-ADDR0 (see Figure 2). These bits can only be changed when ADCI and ADCS are both low.

The completion of the 8-bit ADC conversion is flagged by ADCI in the ADCON register and the result is stored in the special function register ADAT.

An ADC conversion in progress is unaffected by a software ADC start. The result of a completed conversion remains unaffected provided ADCI remains at a logic 1. While ADCS is a logic 1 or ADCI is a logic 1, a new ADC START will be blocked and consequently lost. An A/D conversion in progress will be aborted when the idle or power-down mode is entered. The result of a

completed conversion (ADCI = logic 1) remains unaffected when entering the idle mode, but will be lost if power-down mode is entered. See Figure 3 for the A/D input equivalent circuit.

The analog input pins ADC0-ADC7 may still be used as digital inputs. The analog input channel that is selected by the ADDR2-ADDR0 bits in ADCON cannot be used as a digital input. Reading the selected A/D channel as a digital input will always return a 1. The unselected A/D inputs may always be used as digital inputs.

On RESET the A/D port pins are set to the Digital mode and will work as a normal port and need no further initialization. To use the A/D converter a single byte should be written to ADCON which selects the A/D mux and concurrently sets the ADCS bit to start the A/D conversion. The 40 machine cycles of the A/D conversion include time for signal settling after the mux is selected and before the Sample and Hold procedure is completed.

The circuitry which disables the digital buffer from the port pin is updated at the start of an A/D conversion by setting the ADCS bit in ADCON. After powerup, problems will occur the first time that ADCON is written to if ADCS is not set; in this case, the digital signal disable registers contain random data and some of the 8 port pins will have their digital buffers disabled. When read, these disabled buffers will ignore their input and only return a 1. This condition will be corrected by writing a 1 to ADCS in ADCON which starts and A/D conversion.

Thus, there are two operating modes:

1. DIGITAL ONLY - No Analog inputs are used and ADCON is never written to. In this case pins ADC0-ADC7 are configured as digital inputs.
2. A/D CONVERTER USED - The input multiplexer select field must be written to and ADCS must be set in ADCON. This allows unselected A/D inputs to be used as digital inputs.

## ADCON Register

MSB								LSB							
X	X	X	ADCI	ADCS	AADR2	AADR1	AADR0								

ADCI	ADCS	Operation
0	0	ADC not busy, a conversion can be started.
0	1	ADC busy, start of a new conversion is blocked.
1	0	Conversion completed, start of a new is blocked.
1	1	Not possible.

INPUT CHANNEL SELECTION			
ADDR2	ADDR1	ADDR0	INPUT PIN
0	0	0	P1.0
0	0	1	P1.1
0	1	0	P1.2
0	1	1	P1.3
1	0	0	P1.4
1	0	1	P1.5
1	1	0	P1.6*
1	1	1	P1.7*

\*Not present on 40-pin DIP versions.





## CMOS single-chip 8-bit microcontroller with A/D and watchdog timer

80C550/83C550/87C550

### A/D CONVERTER PARAMETER DEFINITIONS

The following definitions are included to clarify some specifications given and do not represent a complete set of A/D parameter definitions.

#### Absolute Accuracy Error

Absolute accuracy error of a given output is the difference between the theoretical analog input voltage to produce a given output and the actual analog input voltage required to produce the same code. Since the same output code is produced by a band of input voltages, the "required input voltage" is defined as the midpoint of the band of input voltage that will produce that code. Absolute accuracy error not specified with a code is the maximum over all codes.

#### Nonlinearity

If a straight line is drawn between the end points of the actual converter characteristics such that zero offset and full scale errors are removed, then non-linearity is the maximum deviation of the code transitions of the actual characteristics from that of the straight line so constructed. This is also referred to as relative accuracy and also integral non-linearity.

#### Differential Non-Linearity

Differential non-linearity is the maximum difference between the actual and ideal code widths to the converter. The code widths are the differences expressed in LSB between the code transition points, as the input voltage is varied through the range for the complete set of codes.

#### Gain Error

Gain error is the deviation between the ideal and actual analog input voltage required to cause the final code transition to a full-scale output code after the offset error has been removed. This may sometimes be referred to as full scale error.

#### Offset Error

Offset error is the difference between the actual input voltage that causes the first code transition and the ideal value to cause the first code transition. This ideal value is 1/2 LSB above  $V_{ref-}$ .

#### Channel to Channel Matching

Channel to channel matching is the maximum difference between the corresponding code transitions of the actual characteristics taken from different channels under the same temperature, voltage and frequency conditions.

### Crosstalk

Crosstalk is the measured level of a signal at the output of the converter resulting from a signal applied to one deselected channel.

### Total Error

Maximum deviation of any step point from a line connecting the ideal first transition point to the ideal last transition point.

### Relative Accuracy

Relative accuracy error is the deviation of the ADC's actual code transition points from the ideal code transition points on a straight line which connects the ideal first code transition point and the final code transition point, after nullifying offset error and gain error. It is generally expressed in LSBs or in percent of FSR.

### WATCHDOG TIMER

The purpose of the watchdog timer is to reset the microcontroller within a reasonable amount of time if it enters an erroneous state, possibly due to a programming error, electrical noise, or RFI. When enabled, the watchdog circuit will generate a system reset if the user program fails to "feed" (or reload) the watchdog within a predetermined amount of time.

The watchdog timer implemented on the 8XC550 has a programmable interval and can thus be fine tuned to a particular application. If the watchdog function is not used, the timer may still be used as a versatile general purpose timer.

The watchdog function consists of a programmable 13-bit prescaler, and an 8-bit main timer. The main timer is clocked by a tap taken from one of the top 8 bits of the prescaler. The prescaler is incremented once every machine cycle, or 1/12 of the oscillator frequency. Thus, the main counter can be clocked as often as once every 64 machine cycles or as seldom as once every 8192 machine cycles.

When clocked, the main counter decrements. If the main watchdog counter reaches zero, a system reset will occur. To prevent the watchdog timer from under-flowing, the watchdog must be fed before it counts down to zero. When the watchdog is fed, the contents of the WDL register are loaded into the main watchdog counter and the prescaler is cleared.

### WDCON Register

MSB					LSB		
PRE2	PRE1	PRE0	X	X	WDRUN	WDTOF	WDMOD

Symbol	Position	Function
WDCON.7	PRE2	Prescaler select (read/write).
WDCON.6	PRE1	These bits select the prescaler divide ratio according to the following table:
WDCON.5	PRE0	

PRE2	PRE1	PRE0	DIVISOR (FROM $f_{osc}$ )
0	0	0	12 × 64
0	0	1	12 × 64 × 2
0	1	0	12 × 64 × 4
0	1	1	12 × 64 × 8
1	0	0	12 × 64 × 16
1	0	1	12 × 64 × 32
1	1	0	12 × 64 × 64
1	1	1	12 × 64 × 128

WDCON.4	Not used
WDCON.3	Not used
WDCON.2	WDRUN Run control (read/write). This bit turns the timer on (WDRUN = 1) or off (WDRUN = 0) if the timer mode has been selected.
WDCON.1	WDTOF Timeout flag (read/write). This bit is set when the watchdog timer underflows. It is cleared by an external reset and can be cleared by software.
WDCON.0	WDMOD Mode selection (read/write). When WDMOD = 1, the watchdog is selected; when WDMOD = 0, the timer is selected. Selecting the watchdog mode automatically disables power-down mode. WDMOD is cleared by external reset. Once the watchdog mode is selected, this bit can only be cleared by writing a 0 to this bit and then performing a feed operation.

A very specific sequence of events must take place to feed the watchdog timer; it cannot be fed accidentally by a runaway program. The

## CMOS single-chip 8-bit microcontroller with A/D and watchdog timer

80C550/83C550/87C550

following routines demonstrate setting up and feeding the watchdog timer. These routines apply to all versions of the 8XC550 except the ROM part when running from internal program memory.

This routine sets up and starts the watchdog timer. This is not necessary for internal ROM operation, because setup of the watchdog timer on masked ROM parts is accomplished directly via ROM mask options.

```
SetWD: MOV WDL,#0FFh ;Set watch-
      ;dog
      ;reload value.
      MOV WDCON,#0E5;Set up timer.
      ;prescaler,
      ;mode, and
      ;run bits.
      ACALL FeedWD ;Start watch-
      ;dog with a
      ;feed
      ;operation.
```

RET

This routine executes a watchdog timer feed operation, causing the timer to reload from WDL. Interrupts must be disabled during this operation due to the fact that the two feed registers must be loaded on consecutive instruction cycles, or a system reset will occur immediately.

```
FeedWD: CLR EA ;This
      ;sequence
      ;must not be
      ;interrupted.
      MOV WFEED1,#0A5h;First
      ;instruction
      ;of feed
      ;sequence.
      MOV WFEED2,#05Ah;Second
      ;instruction
      ;of feed
      ;sequence.
      SETB EA ;Turn
      ;interrupts
      ;back on.
      RET
```

An interrupt is available to allow the watchdog timer to be used as a general purpose timer in applications where the watchdog function is not needed. The timer operates in the same manner when used as a general purpose timer except that the timer interrupt is generated on timer underflow instead of a chip reset. Refer to the 87C550 data sheet for additional information on watchdog timer operation.

### Programming the Watchdog Timer

Both the EPROM and ROM devices have a set of SFRs for holding the watchdog autoload values and the control bits. The watchdog time-out flag is present in the watchdog control register and operates the

same in all versions. In the EPROM device, the watchdog parameters (autoload value and control) are always taken from the SFRs. In the ROM device, the watchdog parameters can be mask programmed or taken from the SFRs. The selection to take the watchdog parameters from the SFRs or from the mask programmed values is controlled by EA (external access). When EA is high (internal ROM access), the watchdog parameters are taken from the mask programmed values. If the watchdog is masked programmed to the timer mode, then the autoload values and the pre-scaler taps are taken from the SFRs. When EA is low (external access), the watchdog parameters are taken from the SFRs. The user should be able to leave code in his program which initializes the watchdog SFRs even though he has migrated to the mask ROM part. This allows no code changes from EPROM prototyping to ROM coded production parts.

### Watchdog Detailed Operation

#### EPROM Device

##### (and ROMless Operation: EA = 0)

In the ROMless operation (ROM part, EA = 0) and in the EPROM device, the watchdog operates in the following manner.

Whether the watchdog is in the watchdog or timer mode, when external RESET is applied, the following takes place:

- Watchdog mode bit set to timer mode.
- Watchdog run control bit set to OFF.
- Autoload register set to FF (max count).
- Watchdog time-out flag cleared.
- Prescaler is cleared.
- Prescaler tap set to the highest divide.
- Autoload takes place.

The watchdog can be fed even though it is in the timer mode.

Note that the operational concept is for the watchdog mode of operation, when coming out of a hardware reset, the software should load the autoload registers, set the mode to watchdog, and then feed the watchdog (cause an autoload). The watchdog will now be starting at a known point.

If the watchdog is in the watchdog mode and running and happens to underflow at the time the external RESET is applied, the watchdog time-out flag will be cleared.

When the watchdog is in the watchdog mode and the watchdog underflows, the following action takes place:

- Autoload takes place.
- Watchdog time-out flag is set

- Timer mode interrupt flag unchanged.
- Mode bit unchanged.
- Watchdog run bit unchanged.
- Autoload register unchanged.
- Prescaler tap unchanged.
- All other device action same as external reset.

Note that if the watchdog underflows, the program counter will start from 00H as in the case of an external reset. The watchdog time-out flag can be examined to determine if the watchdog has caused the reset condition. The watchdog time-out flag bit can be cleared by software.

When the watchdog is in the timer mode and the timer software underflows, the following action takes place:

- Autoload takes place.
- Watchdog time-out flag is set
- Mode bit unchanged.
- Watchdog run bit unchanged.
- Autoload register unchanged.
- Prescaler tap unchanged.

The timer mode interrupt flag is cleared when the interrupt routine is invoked. This bit can also be cleared directly by software without a software feed operation.

#### Mask ROM Device (EA = 1)

In the mask ROM device, the watchdog mode bit (WDMOD) is mask programmed and the bit in the watchdog command register is read only and reflects the mask programmed selection. If the mask programmed mode bit selects the timer mode, then the watchdog run bit (WDRUN) operates as described under EPROM Device. If the mask programmed bit selects the watchdog mode, then the watchdog run bit has no effect on the timer operation.

#### Watchdog Function

The watchdog consists of a programmable prescaler and the main timer. The prescaler derives its clock from the on-chip oscillator. The prescaler consists of a divide by 12 followed by a 13 stage counter with taps from stage 6 through stage 13. The tap selection is programmable. The watchdog main counter is a down counter clocked (decremented) each time the programmable prescaler underflows. The watchdog generates an underflow signal (and is autoloaded) when the watchdog is at count 0 and the clock to decrement the watchdog occurs. The watchdog is 8 bits long and the autoload

## CMOS single-chip 8-bit microcontroller with A/D and watchdog timer

## 80C550/83C550/87C550

value can range from 0 to FFH. (The autoloader value of 0 is permissible since the prescaler is cleared upon autoloader.)

This leads to the following user design equations. Definitions:  $t_{OSC}$  is the oscillator period,  $N$  is the selected prescaler tap value,  $W$  is the main counter autoloader value,  $t_{MIN}$  is the minimum watchdog time-out value (when the autoloader value is 0),  $t_{MAX}$  is the maximum time-out value (when the autoloader value is FFH),  $t_D$  is the design time-out value.

$$t_{MIN} = t_{OSC} \times 12 \times 64$$

$$t_{MAX} = t_{MIN} \times 128 \times 256$$

$$t_D = t_{MIN} \times 2^{PRESCALER} \times W$$

(where prescaler = 0, 1, 2, 3, 4, 5, 6, or 7)

Note that the design procedure is anticipated to be as follows. A  $t_{MAX}$  will be chosen either from equipment or operation considerations and will most likely be the next convenient value higher than  $t_D$ . (If the watchdog were inadvertently to start from FFH, an overflow would be guaranteed, barring other anomalies, to occur within  $t_{MAX}$ ). Then the value for the prescaler would be chosen from:

$$\text{prescaler} = \log_2 (t_{MAX} / (t_{OSC} \times 12 \times 256)) - 6$$

This then also fixes  $t_{MIN}$ . An autoloader value would then be chosen from:

$$W = t_D / t_{MIN} - 1$$

The software must be written so that a feed operation takes place every  $t_D$  seconds from the last feed operation. Some tradeoffs may need to be made. It is not advisable to include feed operations in minor loops or in subroutines unless the feed operation is a specific subroutine.

### Interrupts

The 8XC550 interrupt structure is a seven-source, two-priority level interrupt system similar to that of the standard 80C51 microcontroller. The interrupt sources are listed below in the order of their internal polling sequence. This is the order in which simultaneous interrupts of the same priority level would be serviced.

### Interrupt Priorities

PRIORITY	SOURCE	VECTOR ADDRESS	FUNCTION
Highest	INT0	0003H	External interrupt 0
	TF0	000BH	Counter/timer 0 overflow
	INT1	0013H	External interrupt 1
	TF1	001BH	Counter/timer 1 overflow
	TI & RI	0023H	Serial port transmit/receive
Lowest	ADCI	002BH	A/D converter conversion complete
	WDTOF	0033H	Watchdog timer overflow (only when not in watchdog mode)

### Interrupt Control Registers

The standard 80C51 interrupt enable and priority registers have been modified slightly to take into account the additional interrupt sources of the 8XC550.

### Interrupt Enable Register

MSB				LSB			
EA	EWD	EAD	ES	ET1	EX1	ET0	EX0
Symbol	Position	Function					
EA	IE.7	Global interrupt enable					
EWD	IE.6	Watchdog timer overflow					
EAD	IE.5	A/D conversion complete					
ES	IE.4	Serial port transmit or receive					
ET1	IE.3	Timer 1 overflow					
EX1	IE.2	External interrupt 1					
ET0	IE.1	Timer 0 overflow					
EX0	IE.0	External interrupt 0					

### Interrupt Priority Register

MSB				LSB			
-	PWD	PAD	PS	PT1	PT0	PX0	PX0
Symbol	Position	Function					
PWD	IP.6	Watchdog timer					
PAD	IP.5	A/D conversion					
PS	IP.4	Serial port interrupt					
PT1	IP.3	Timer 1 interrupt					
PX1	IP.2	External interrupt 1					
PT0	IP.1	Timer 0 interrupt					
PX0	IP.0	External interrupt 0					

### Power-Down and Idle Modes

The 8XC550 includes the standard 80C51 power-down and idle modes of reduced power consumption. In addition, the 8XC550 includes an option to separately turn off the serial port for extra power savings when it is not needed. Also, the individual functional blocks such as the counter/timers are automatically disabled when they are not running. This actually turns off the clocks to the block in question, resulting in additional power savings. Note that when the watchdog timer is operating, the processor is inhibited from entering the power-down mode. This is due to the fact that the oscillator is stopped in the power-down mode, which would effectively turn off the watchdog timer. In keeping with the purpose of the watchdog timer, the processor is prevented from accidentally entering power-down due to some erroneous operation.

### Power Control Register

MSB				LSB			
SMOD	SIDL	-	-	GF1	GF0	PD	IDL

Symbol	Position	Function
SMOD	PCON.7	Double baud rate bit. When set to a 1 and Timer 1 is used to generate baud rate, and the serial port is used in modes 1, 2, or 3.

Symbol	Position	Function
SIDL	PCON.6	Separately idles the serial port for additional power savings.
-	PCON.5	Reserved
-	PCON.4	Reserved
GF1	PCON.3	General-purpose flag bit.
GF0	PCON.2	General-purpose flag bit.
PD	PCON.1	Power-down bit. Starting this bit activates power-down operation.
IDL	PCON.0	Idle mode bit. Setting this bit activates idle mode operation.

If 1s are written to PD and IDL at the same time, PD takes precedence.

CMOS single-chip 8-bit microcontroller

with A/D and watchdog timer

80C550/83C550/87C550

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator, as shown in the Block Diagram, page 555).

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

IDLE MODE

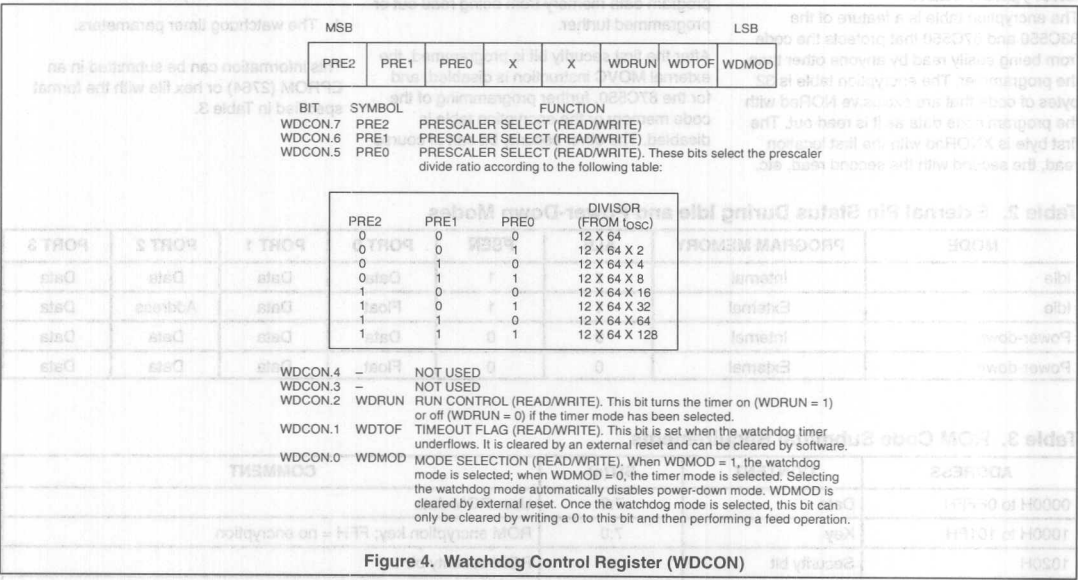
In idle mode, the CPU puts itself to sleep while all of the on-chip peripherals except the A/D stay active. the instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. An A/D conversion in progress will be aborted when idle mode is entered. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

Programmable Idle Modes

The programmable idle modes have been dispersed throughout the functional blocks. Each block has its own ability to be disabled. For example, if timer 0 is not commanded to be running (TR = 0), then the clock to the timer is disabled resulting in an idle mode power saving. An additional idle control bit has been added to the serial communications port.

A/D Operation in Idle Mode

When in the idle mode, the A/D converter will be disabled. However, the current through the VREF pins will be present and will not be reduced internally in either the idle or the power-down modes. It is the responsibility of the user to disconnect VREF to reduce power supply current.





# CMOS single-chip 8-bit microcontroller with A/D and watchdog timer

80C550/83C550/87C550

## DESIGN CONSIDERATIONS

At power-on, the voltage on V<sub>CC</sub> and RST must come up at the same time for a proper start-up.

When the idle mode is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write when idle is terminated by reset, the instruction following the one that invokes idle should not be one that writes to a port pin or to external memory. Table 2 shows the state of I/O ports during low current operating modes.

## Encryption Table

The encryption table is a feature of the 83C550 and 87C550 that protects the code from being easily read by anyone other than the programmer. The encryption table is 32 bytes of code that are exclusive NORed with the program code data as it is read out. The first byte is XNORed with the first location read, the second with the second read, etc.

After the encryption table has been programmed, the user has to know its contents in order to correctly decode the program code data. The encryption table itself cannot be read out.

For the EPROM (87C550) part, the encryption table is programmed in the same manner as the program memory, but using the "Pgm Encryption Table" levels specified in Table 4. After the encryption table is programmed, verification cycles will produce only encrypted information.

For the ROM part (83C550) the encryption table information is submitted with the ROM code as shown in Table 3.

## Security Bits

There are two security bits on the 83C550 and 87C550 that, when set, prevent the program data memory from being read out or programmed further.

After the first security bit is programmed, the external MOV<sub>C</sub> instruction is disabled, and for the 87C550, further programming of the code memory or the encryption table is disabled. The other security bit can of course

still be programmed. With only security bit one programmed, the memory can still be read out for program verification. After the second security bit is programmed, it is no longer possible to read out (verify) the program memory.

To program the security bits for the 87C550, repeat the programming sequence using the "Pgm Security Bit" levels specified in Table 4. For the masked ROM 83C550 the security bit information is submitted with the ROM code as shown in Table 3.

## ROM Code Submission

When submitting a ROM code for the 83C550, the following must be specified:

1. The 4k byte user ROM program.
2. The 32 byte ROM encryption key.
3. The ROM security bits.
4. The watchdog timer parameters.

This information can be submitted in an EPROM (2764) or hex file with the format specified in Table 3.

Table 2. External Pin Status During Idle and Power-Down Modes

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

Table 3. ROM Code Submittal Requirements

ADDRESS	CONTENT	BIT(s)	COMMENT
0000H to 0FFFFH	Data	7:0	User ROM data
1000H to 101FH	Key	7:0	ROM encryption key; FFH = no encryption
1020H	Security bit	0	ROM security bit 1
1020H	Security bit	1	ROM security bit 2 0 = enable security feature 1 = disable security feature
1030H	WMOD	0	Watchdog mode bit; 00H = timer mode 01H = watchdog mode
1031H	PRE2:0	2:0	Watchdog prescaler selection; 00H = divide by 12 x 64 07H = divide by 12 x 64 x 128 (see specification)
1032H	WD	7:0	Watchdog autoloading value (see specification)



# CMOS single-chip 8-bit microcontroller with A/D and watchdog timer

80C550/83C550/87C550

**Electrical Deviations from Commercial Specifications for Extended Temperature Range**  
DC and AC parameters not included here are the same as in the commercial temperature range table.

## DC ELECTRICAL CHARACTERISTICS

$T_{amb} = -40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$  (87C550),  $V_{CC} = 5\text{V} \pm 20\%$  (80/83C550),  $V_{SS} = 0\text{V}$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			MIN	MAX	
$V_{IL}$	Input low voltage, except EA		-0.5	$0.2V_{CC} - 0.15$	V
$V_{IL1}$	Input low voltage to EA		0	$0.2V_{CC} - 0.35$	V
$V_{IH}$	Input high voltage, except XTAL1, RST		$0.2V_{CC} + 1$	$V_{CC} + 0.5$	V
$V_{IH1}$	Input high voltage to XTAL1, RST		$0.7V_{CC} + 0.1$	$V_{CC} + 0.5$	V
$I_{IL}$	Logical 0 input current, ports 2, 3	$V_{IN} = 0.45\text{V}$		-75	$\mu\text{A}$
$I_{TL}$	Logical 1-to-0 transition current, ports 2, 3	$V_{IN} = 2.0\text{V}$		-750	$\mu\text{A}$
$I_{CC}$	Power supply current: Active mode Idle mode Power down mode	$V_{CC} = 4.5\text{--}5.5\text{V}$ , Frequency range = 3.5 to 16MHz		35 6 50	mA mA $\mu\text{A}$

## ADC DC ELECTRICAL CHARACTERISTICS

$AV_{CC} = 5\text{V} \pm 10\%$ ,  $AV_{SS} = 0\text{V}$ ,  $T_{amb} = -40^{\circ}\text{C}$  to  $85^{\circ}\text{C}$ , unless otherwise specified

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			MIN	MAX	
$AV_{CC}$	Analog supply	$AV_{CC} = V_{CC} \pm 0.2$	4.5	5.5	V
$V_{REF}$	Analog reference; $AV_{REF+}$ $AV_{REF-}$		$AV_{SS} - 0.2$	$AV_{CC} + 0.2$	V
$AI_{CC}$	Analog operating supply current	See note 1		3.0	mA
$AV_{IN}$	Analog input voltage		$AV_{SS} - 0.2$	$AV_{CC} + 0.2$	V
$A_{IC}, C_{IA}$	Analog input capacitance			15	pF
$t_{ADS}$	Sampling time			$8t_{CY}$	
$t_{ADC}$	Conversion time			$40t_{CY}$	
$A_e$	Absolute voltage error			$\pm 1.5$	LSB
$E_{RA}$	Relative accuracy			$\pm 1$	LSB
$OS_e$	Offset error	See note 1		$\pm 1$	LSB
$Ge$	Gain error	See note 1		0.4	%
$M_{CTC}$	Channel-to-channel matching			$\pm 1$	LSB
$Ct_{Am}$	Crosstalk	0 – 100kHz		-60	dB
$R_{ref}$	Resistance between $AV_{REF+}$ and $AV_{REF-}$		1.0	10.0	$K\Omega$
$AI_{ID}$	Idle mode supply current	See note 4		50	$\mu\text{A}$
$AI_{PD}$	Power down supply current	See note 4		50	$\mu\text{A}$

### NOTES:

- Conditions:  $V_{REF+} = 4.99712\text{V}$ ,  $V_{REF-} = 0\text{V}$ .  $AI_{CC}$  value does not include the resistor ladder current. For the 40-pin package, where the  $V_{REF-}$  inputs are connected to  $AV_{CC}$  and  $AV_{SS}$ , the current  $AI_{CC}$  will be increased by the resistor ladder current and may exceed the maximum shown here.
- The resistor ladder network is not disconnected in the power-down or idle modes. Thus to conserve power, the user must remove  $AV_{CC}$  and  $V_{REF+}$ .
- If the A/D function is not required, or if the A/D function is only needed periodically,  $AV_{CC}$  can be removed without affecting the operation of the digital circuitry. Contents of ADCON and ADAT are not guaranteed to be valid. Digital inputs P1.0 to P1.7 will not function normally. No digital outputs are present on these pins.
- For this test, the Analog inputs must be at the supplies (either  $V_{DD}$  or  $V_{SS}$ ).

**ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>**

PARAMETER	RATING	UNIT
Operating temperature under bias	-40 to +85	°C
Storage temperature range	-65 to +150	°C
Voltage on EA/V <sub>PP</sub> pin to V <sub>SS</sub> (87C550 only)	0 to +13.0	V
Voltage on any other pin to V <sub>SS</sub>	-0.5 to +6.5	V
Input, output current on any two I/O pins	±10	mA
Power dissipation (based on package heat transfer limitations, not device power consumption)	1.5	W

**NOTES:**

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
- This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
- Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V<sub>SS</sub> unless otherwise noted.

**DC ELECTRICAL CHARACTERISTICS**

T<sub>amb</sub> = 0°C to +70°C or -40°C to +85°C, V<sub>CC</sub> = 5V ±10% (87C550), V<sub>CC</sub> = 5V ±20% (80/83C550), V<sub>SS</sub> = 0V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYPICAL <sup>1</sup>	MAX	
V <sub>IL</sub>	Input low voltage, except EA <sup>7</sup>		-0.5		0.2V <sub>CC</sub> -0.1	V
V <sub>IL1</sub>	Input low voltage to EA <sup>7</sup>		0		0.2V <sub>CC</sub> -0.3	V
V <sub>IH</sub>	Input high voltage, except XTAL1, RST <sup>7</sup>		0.2V <sub>CC</sub> +0.9		V <sub>CC</sub> +0.5	V
V <sub>IH1</sub>	Input high voltage, XTAL1, RST <sup>7</sup>		0.7V <sub>CC</sub>		V <sub>CC</sub> +0.5	V
V <sub>OL</sub>	Output low voltage, ports 2, 3	I <sub>OL</sub> = 1.6mA <sup>2</sup>			0.45	V
V <sub>OL1</sub>	Output low voltage, port 0, ALE, PSEN	I <sub>OL</sub> = 3.2mA <sup>2</sup>			0.45	V
V <sub>OH</sub>	Output high voltage, ports 2, 3, ALE, PSEN <sup>3</sup>	I <sub>OH</sub> = -60μA, I <sub>OH</sub> = -25μA, I <sub>OH</sub> = -10μA	2.4 0.75V <sub>CC</sub> 0.9V <sub>CC</sub>			V V V
V <sub>OH1</sub>	Output high voltage (port 0 in external bus mode)	I <sub>OH</sub> = -800μA, I <sub>OH</sub> = -300μA, I <sub>OH</sub> = -80μA	2.4 0.75V <sub>CC</sub> 0.9V <sub>CC</sub>			V V V
I <sub>IL</sub>	Logical 0 input current, ports 1, 2, 3 <sup>7</sup>	V <sub>IN</sub> = 0.45V			-50	μA
I <sub>TL</sub>	Logical 1-to-0 transition current, ports 1, 2, 3 <sup>7</sup>	See note 4			-650	μA
I <sub>LI</sub>	Input leakage current, port 0	V <sub>IN</sub> = V <sub>IL</sub> or V <sub>IH</sub>			±10	μA
I <sub>CC</sub>	Power supply current (does not include A <sub>I</sub> CC): <sup>7</sup> Active mode @ 16MHz <sup>5</sup> Idle mode @ 16MHz Power down mode	See note 6		11.5 1.3 3	25 5 50	mA mA μA
R <sub>RST</sub>	Internal reset pull-down resistor		50		300	kΩ
C <sub>IO</sub>	Pin capacitance (I/O pins only)				10	pF

**NOTES:**

- Typical ratings are not guaranteed. The values listed are at room temperature, 5V.
- Capacitive loading on ports 0 and 2 may cause spurious noise to be superimposed on the V<sub>OL</sub>s of ALE and ports 1 and 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100pF), the noise pulse on the ALE pin may exceed 0.8V. In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.
- Capacitive loading on ports 0 and 2 may cause the V<sub>OH</sub> on ALE and PSEN to momentarily fall below the 0.9V<sub>CC</sub> specification when the address bits are stabilizing.
- Pins of ports 2 and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when V<sub>IN</sub> is approximately 2V.
- I<sub>CC</sub>MAX at other frequencies is given by: Active mode: I<sub>CC</sub>MAX = 1.43 × FREQ + 1.90; Idle mode: I<sub>CC</sub>MAX = 0.14 × FREQ + 2.31, where FREQ is the external oscillator frequency in MHz. I<sub>CC</sub>MAX is given in mA. See Figure 12.
- See Figures 13 through 16 for I<sub>CC</sub> test conditions.
- These values apply only to T<sub>amb</sub> = 0°C to +70°C. For T<sub>amb</sub> = -40°C to +85°C. See table on previous page.

# CMOS single-chip 8-bit microcontroller with A/D and watchdog timer

80C550/83C550/87C550

## AC ELECTRICAL CHARACTERISTICS

 $T_{amb} = 0^{\circ}\text{C to } +70^{\circ}\text{C or } -40^{\circ}\text{C to } +85^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$  (87C550),  $V_{CC} = 5\text{V} \pm 20\%$  (80/83C550),  $V_{SS} = 0\text{V}^{1,2}$ 

SYMBOL	FIGURE	PARAMETER	16MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
$1/t_{CLCL}$	5	Oscillator frequency: <b>Speed Versions</b> S8XC550 Exx			3.5	16	MHz
$t_{LHLL}$	5	ALE pulse width	85		$2t_{CLCL}-40$		ns
$t_{AVLL}$	5	Address valid to ALE low	7		$t_{CLCL}-55$		ns
$t_{LLAX}$	5	Address hold after ALE low	27		$t_{CLCL}-35$		ns
$t_{LLIV}$	5	ALE low to valid instruction in		150		$4t_{CLCL}-100$	ns
$t_{LLPL}$	5	ALE low to PSEN low	22		$t_{CLCL}-40$		ns
$t_{PLPH}$	5	PSEN pulse width	142		$3t_{CLCL}-45$		ns
$t_{PLIV}$	5	PSEN low to valid instruction in		82		$3t_{CLCL}-105$	ns
$t_{PXIX}$	5	Input instruction hold after PSEN	0		0		ns
$t_{PXIZ}$	5	Input instruction float after PSEN		37		$t_{CLCL}-25$	ns
$t_{AVIV}$	5	Address to valid instruction in		207		$5t_{CLCL}-105$	ns
$t_{PLAZ}$	5	PSEN low to address float		10		10	ns
<b>Data Memory</b>							
$t_{RLRH}$	6, 7	RD pulse width	275		$6t_{CLCL}-100$		ns
$t_{WLWH}$	6, 7	WR pulse width	275		$6t_{CLCL}-100$		ns
$t_{RLDV}$	6, 7	RD low to valid data in		212		$5t_{CLCL}-165$	ns
$t_{RHDZ}$	6, 7	Data hold after RD	0		0		ns
$t_{RHDZ}$	6, 7	Data float after RD		55		$2t_{CLCL}-70$	ns
$t_{LLDV}$	6, 7	ALE low to valid data in		350		$8t_{CLCL}-150$	ns
$t_{AVDV}$	6, 7	Address to valid data in		397		$9t_{CLCL}-165$	ns
$t_{LLWL}$	6, 7	ALE low to RD or WR low	137	247	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
$t_{AVWL}$	6, 7	Address valid to WR low or RD low	120		$4t_{CLCL}-130$		ns
$t_{QVWX}$	6, 7	Data valid to WR transition	12		$t_{CLCL}-50$		ns
$t_{WHQX}$	6, 7	Data hold after WR	12		$t_{CLCL}-50$		ns
$t_{RLAZ}$	6, 7	RD low to address float		0		0	ns
$t_{WHLH}$	6, 7	RD or WR high to ALE high	22	102	$t_{CLCL}-40$	$t_{CLCL}+40$	ns
<b>External Clock</b>							
$t_{CHCX}$	9	High time	20		20		ns
$t_{CLCX}$	9	Low time	20		20		ns
$t_{CLCH}$	9	Rise time		20		20	ns
$t_{CHCL}$	9	Fall time		20		20	ns
<b>Shift Register</b>							
$t_{XLXL}$	8	Serial port clock cycle time	750		$12t_{CLCL}$		ns
$t_{QVXH}$	8	Output data setup to clock rising edge	492		$10t_{CLCL}-133$		ns
$t_{XHGX}$	8	Output data hold after clock rising edge	8		$2t_{CLCL}-117$		ns
$t_{XHDZ}$	8	Input data hold after clock rising edge	0		0		ns
$t_{XHDV}$	8	Clock rising edge to input data valid		492		$10t_{CLCL}-133$	ns

### NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.

# CMOS single-chip 8-bit microcontroller with A/D and watchdog timer

80C550/83C550/87C550

## EXPLANATION OF THE AC SYMBOLS

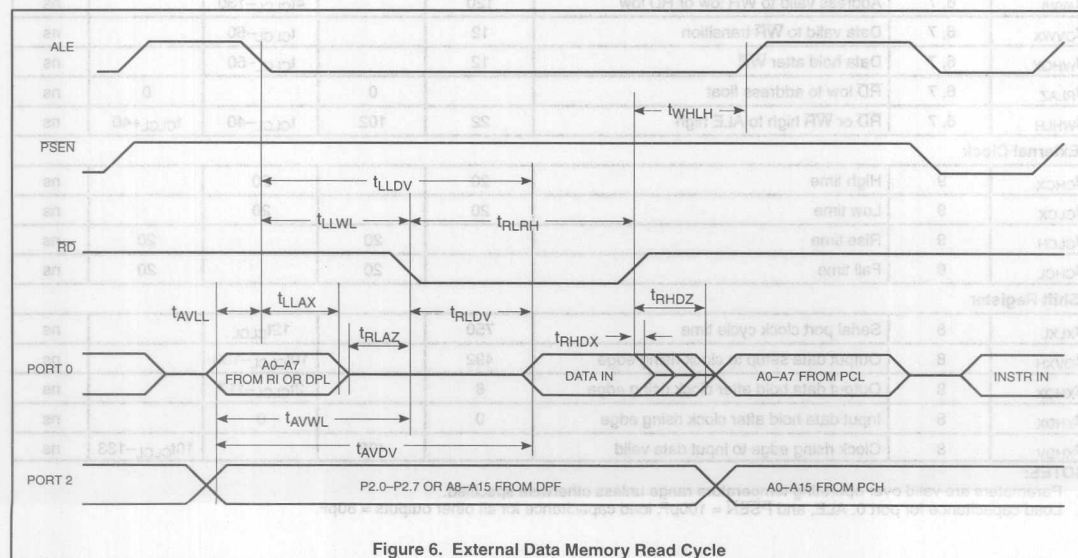
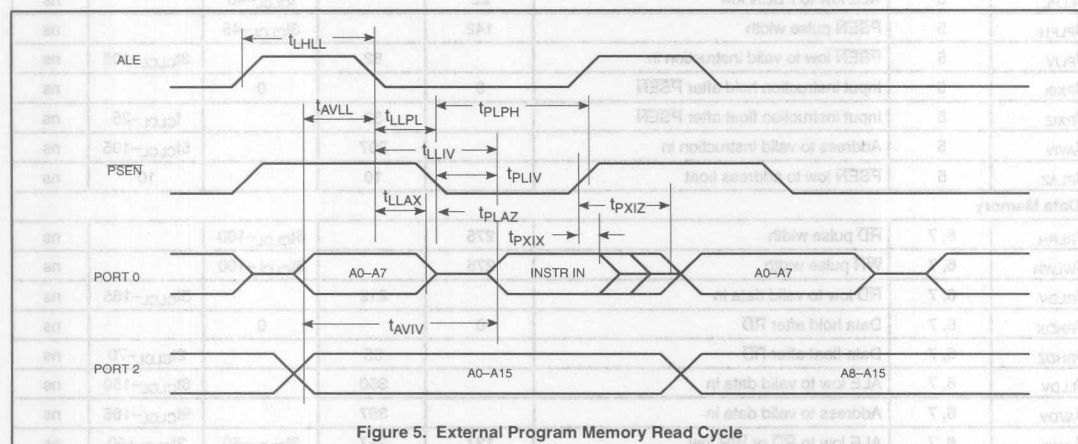
Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

A - Address  
C - Clock  
D - Input data  
H - Logic level high  
I - Instruction (program memory contents)  
L - Logic level low, or ALE  
P - PSEN

Q - Output data  
R - RD signal  
t - Time  
V - Valid  
W - WR signal  
X - No longer a valid logic level  
Z - Float

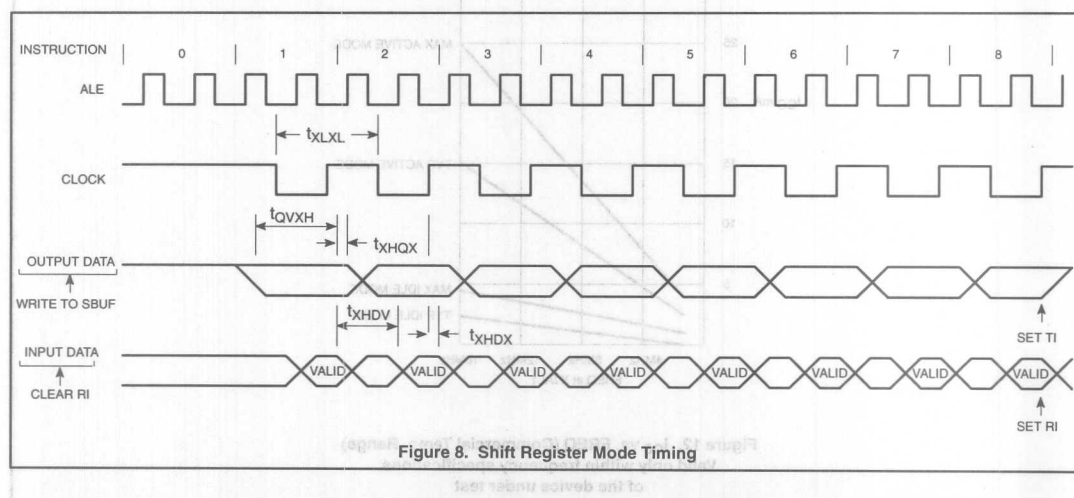
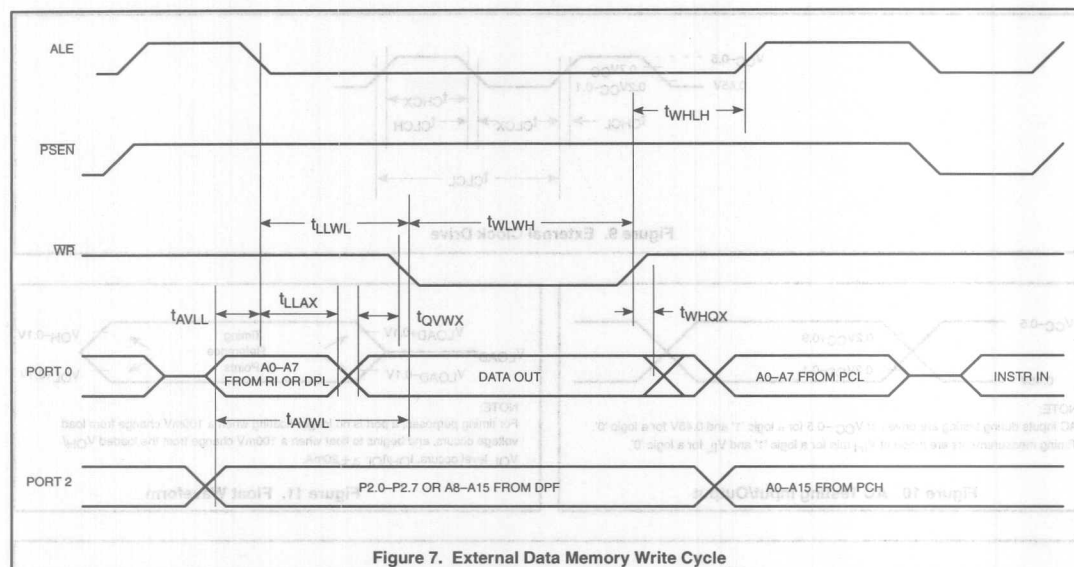
Examples:  $t_{AVLL}$  = Time for address valid to ALE low.

$t_{LLPL}$  = Time for ALE low to PSEN low.



# CMOS single-chip 8-bit microcontroller with A/D and watchdog timer

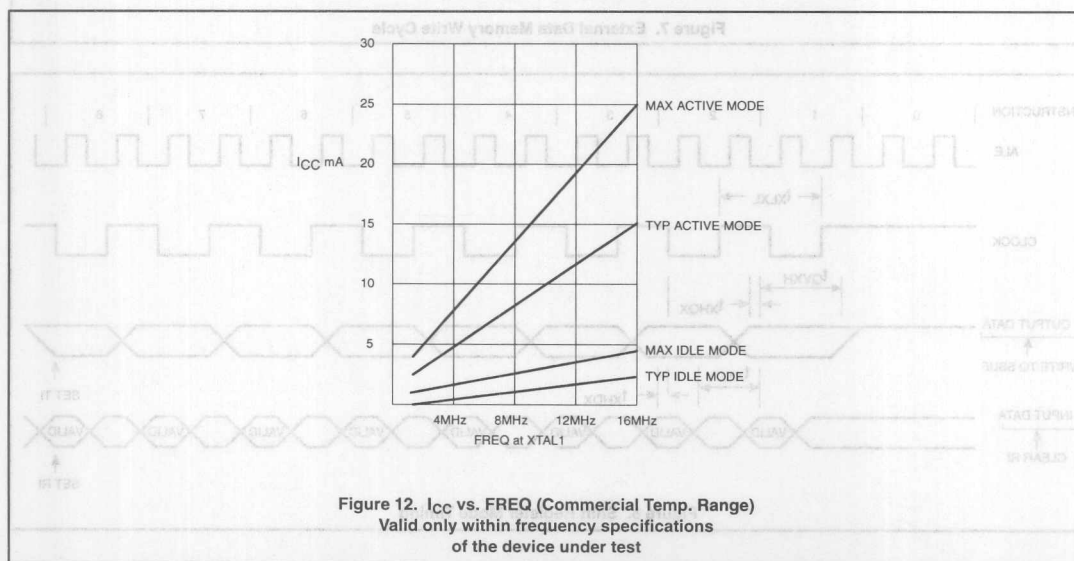
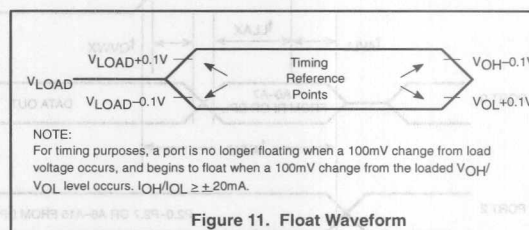
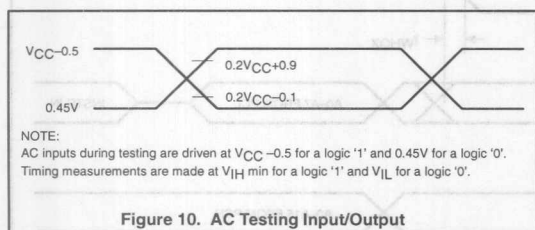
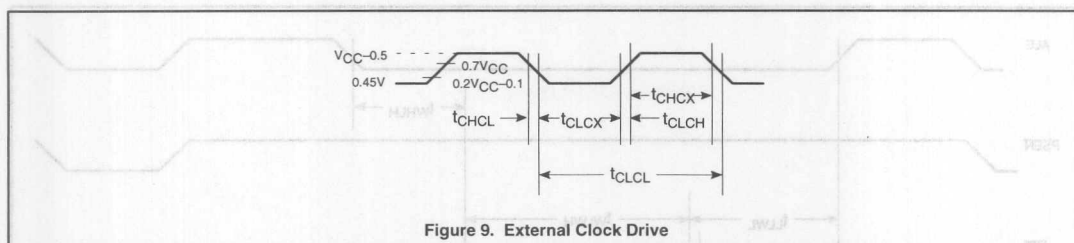
80C550/83C550/87C550





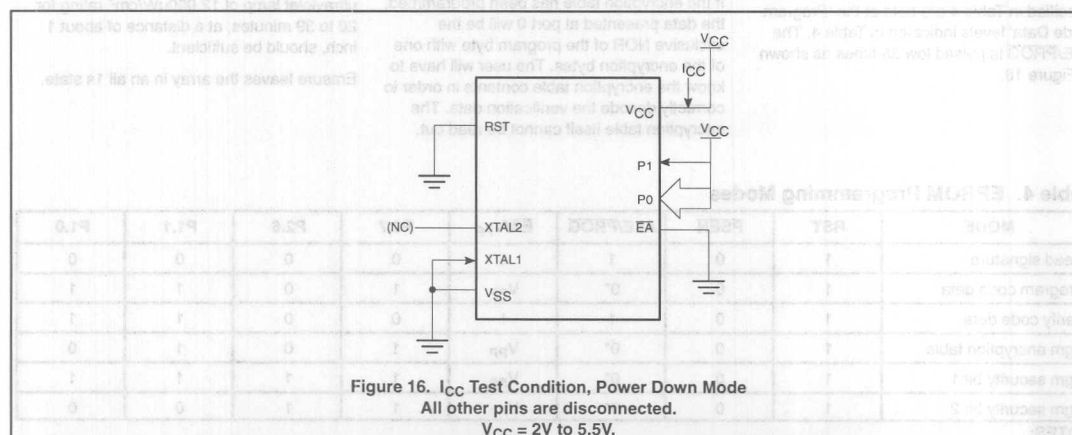
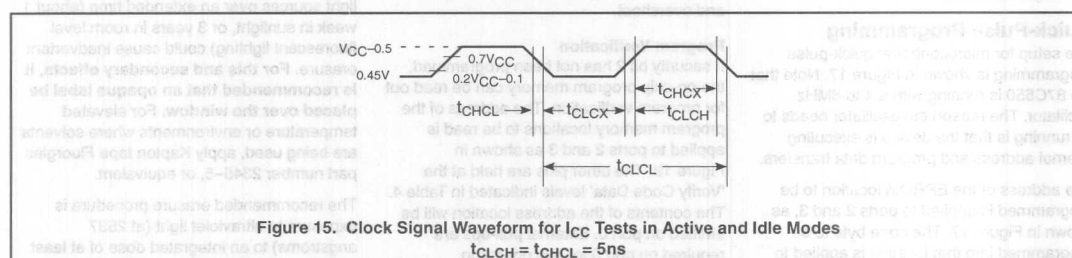
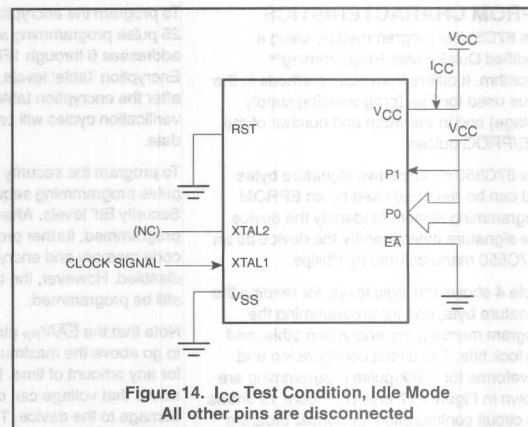
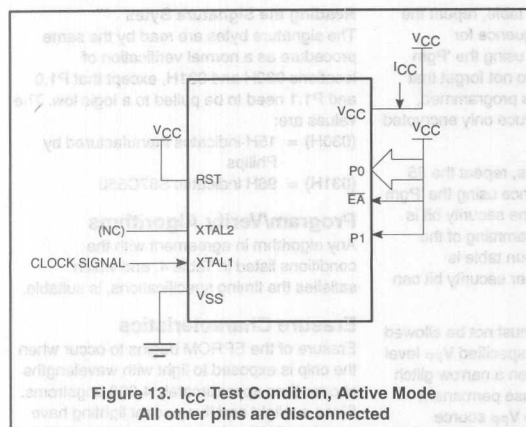
# CMOS single-chip 8-bit microcontroller with A/D and watchdog timer

80C550/83C550/87C550



# CMOS single-chip 8-bit microcontroller with A/D and watchdog timer

## 80C550/83C550/87C550



## CMOS single-chip 8-bit microcontroller with A/D and watchdog timer

80C550/83C550/87C550

### EPROM CHARACTERISTICS

The 87C550 is programmed by using a modified Quick-Pulse Programming™ algorithm. It differs from older methods in the value used for  $V_{PP}$  (programming supply voltage) and in the width and number of the ALE/PROG pulses.

The 87C550 contains two signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes identify the device as an S87C550 manufactured by Philips.

Table 4 shows the logic levels for reading the signature byte, and for programming the program memory, the encryption table, and the lock bits. The circuit configuration and waveforms for quick-pulse programming are shown in Figures 17 and 18. Figure 19 shows the circuit configuration for normal program memory verification.

### Quick-Pulse Programming

The setup for microcontroller quick-pulse programming is shown in Figure 17. Note that the 87C550 is running with a 4 to 6MHz oscillator. The reason the oscillator needs to be running is that the device is executing internal address and program data transfers.

The address of the EPROM location to be programmed is applied to ports 2 and 3, as shown in Figure 17. The code byte to be programmed into that location is applied to port 0. RST, PSEN and pins of ports 1 and 2 specified in Table 4 are held at the 'Program Code Data' levels indicated in Table 4. The ALE/PROG is pulsed low 25 times as shown in Figure 18.

To program the encryption table, repeat the 25 pulse programming sequence for addresses 0 through 1FH, using the 'Pgm Encryption Table' levels. Do not forget that after the encryption table is programmed, verification cycles will produce only encrypted data.

To program the security bits, repeat the 25 pulse programming sequence using the 'Pgm Security Bit' levels. After one security bit is programmed, further programming of the code memory and encryption table is disabled. However, the other security bit can still be programmed.

Note that the  $\overline{EA}/V_{PP}$  pin must not be allowed to go above the maximum specified  $V_{PP}$  level for any amount of time. Even a narrow glitch above that voltage can cause permanent damage to the device. The  $V_{PP}$  source should be well regulated and free of glitches and overshoot.

### Program Verification

If security bit 2 has not been programmed, the on-chip program memory can be read out for program verification. The address of the program memory locations to be read is applied to ports 2 and 3 as shown in Figure 19. The other pins are held at the 'Verify Code Data' levels indicated in Table 4. The contents of the address location will be emitted on port 0. External pull-ups are required on port 0 for this operation.

If the encryption table has been programmed, the data presented at port 0 will be the exclusive NOR of the program byte with one of the encryption bytes. The user will have to know the encryption table contents in order to correctly decode the verification data. The encryption table itself cannot be read out.

### Reading the Signature Bytes

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P1.0 and P1.1 need to be pulled to a logic low. The values are:

(030H) = 15H indicates manufactured by Philips

(031H) = 96H indicates S87C550

### Program/Verify Algorithms

Any algorithm in agreement with the conditions listed in Table 4, and which satisfies the timing specifications, is suitable.

### Erase Characteristics

Erase of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent erasure. **For this and secondary effects, it is recommended that an opaque label be placed over the window.** For elevated temperature or environments where solvents are being used, apply Kapton tape Fluorglas part number 2345-5, or equivalent.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 angstroms) to an integrated dose of at least 15W-sec/cm<sup>2</sup>. Exposing the EPROM to an ultraviolet lamp of 12,000uW/cm<sup>2</sup> rating for 20 to 39 minutes, at a distance of about 1 inch, should be sufficient.

Erase leaves the array in an all 1s state.

Table 4. EPROM Programming Modes

MODE	RST	PSEN	ALE/PROG	$\overline{EA}/V_{PP}$	P2.7	P2.6	P1.1	P1.0
Read signature	1	0	1	1	0	0	0	0
Program code data	1	0	0*	$V_{PP}$	1	0	1	1
Verify code data	1	0	1	1	0	0	1	1
Pgm encryption table	1	0	0*	$V_{PP}$	1	0	1	0
Pgm security bit 1	1	0	0*	$V_{PP}$	1	1	1	1
Pgm security bit 2	1	0	0*	$V_{PP}$	1	1	0	0

#### NOTES:

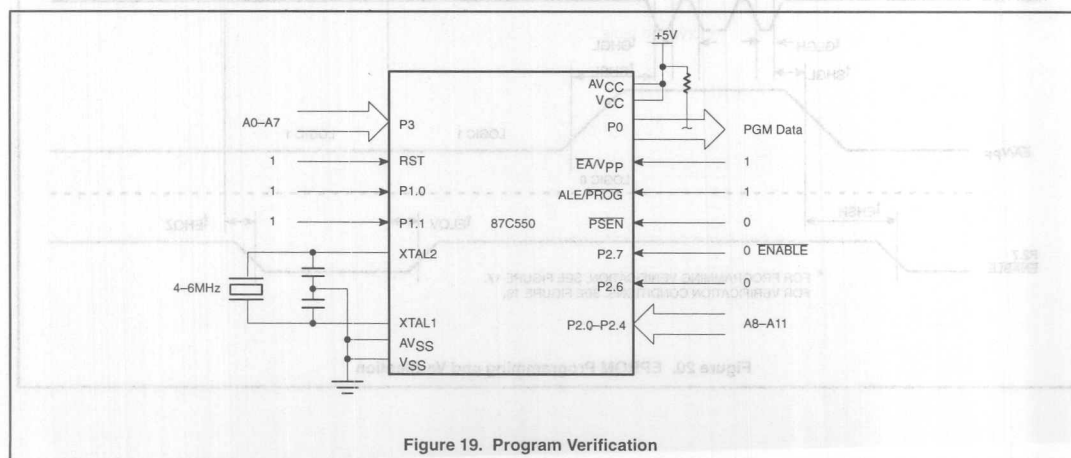
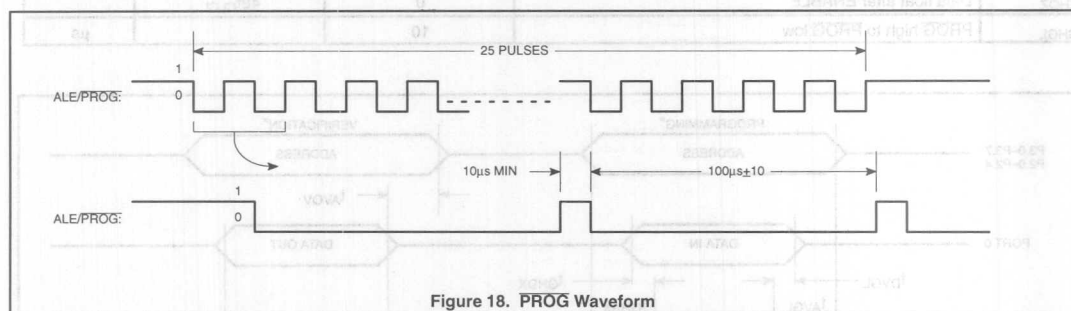
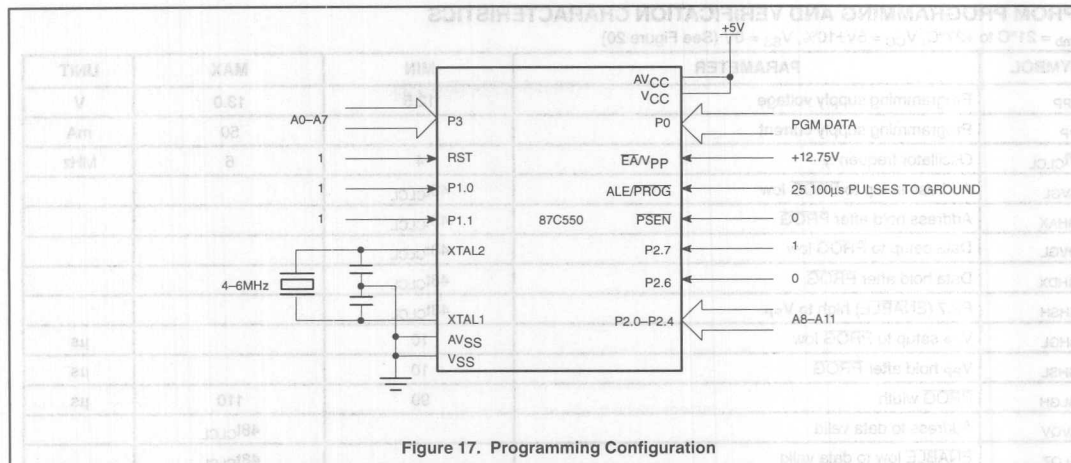
1. '0' = Valid low for that pin, '1' = valid high for that pin.

2.  $V_{PP} = 12.75V \pm 0.25V$ .

3.  $V_{CC} = 5V \pm 10\%$  during programming and verification.

\* ALE/PROG receives 25 programming pulses while  $V_{PP}$  is held at 12.75V. Each programming pulse is low for 100µs (±10µs) and high for a minimum of 10µs.

™Trademark phrase of Intel Corporation.



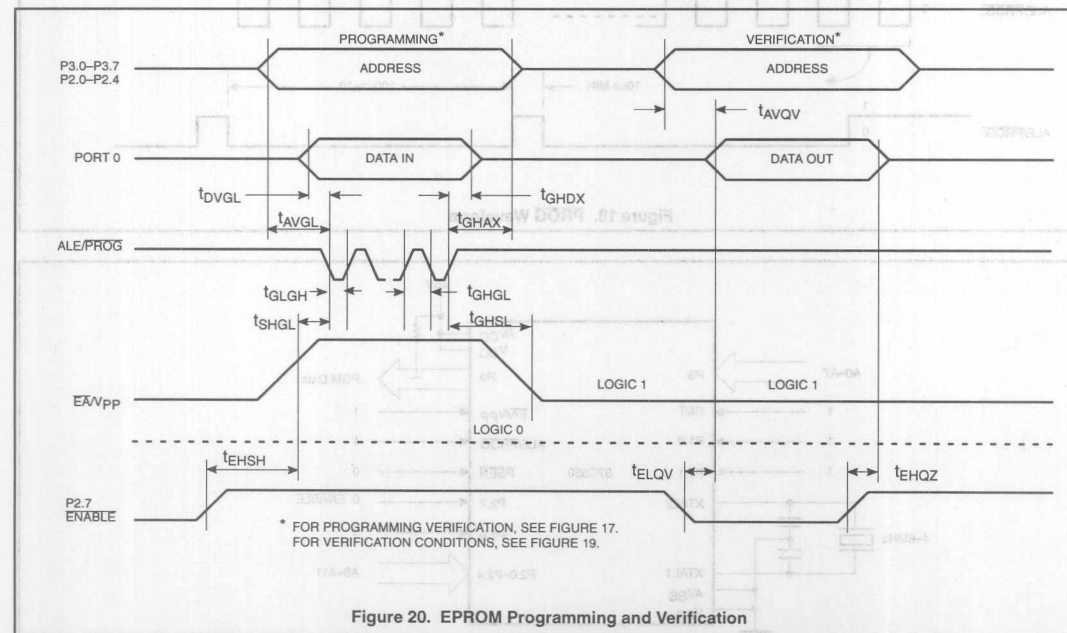
# CMOS single-chip 8-bit microcontroller with A/D and watchdog timer

80C550/83C550/87C550

## EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

 $T_{amb} = 21^{\circ}\text{C}$  to  $+27^{\circ}\text{C}$ ,  $V_{CC} = 5V \pm 10\%$ ,  $V_{SS} = 0V$  (See Figure 20)

SYMBOL	PARAMETER	MIN	MAX	UNIT
$V_{PP}$	Programming supply voltage	12.5	13.0	V
$I_{PP}$	Programming supply current		50	mA
$1/t_{CLCL}$	Oscillator frequency	4	6	MHz
$t_{AVGL}$	Address setup to $\overline{\text{PROG}}$ low	$48t_{CLCL}$		
$t_{GHAX}$	Address hold after $\overline{\text{PROG}}$	$48t_{CLCL}$		
$t_{DVGL}$	Data setup to $\overline{\text{PROG}}$ low	$48t_{CLCL}$		
$t_{GHDX}$	Data hold after $\overline{\text{PROG}}$	$48t_{CLCL}$		
$t_{EHS}$	P2.7 (ENABLE) high to $V_{PP}$	$48t_{CLCL}$		
$t_{SHGL}$	$V_{PP}$ setup to $\overline{\text{PROG}}$ low	10		$\mu\text{s}$
$t_{GHSL}$	$V_{PP}$ hold after $\overline{\text{PROG}}$	10		$\mu\text{s}$
$t_{GLGH}$	$\overline{\text{PROG}}$ width	90	110	$\mu\text{s}$
$t_{AVGV}$	Address to data valid		$48t_{CLCL}$	
$t_{ELOZ}$	ENABLE low to data valid		$48t_{CLCL}$	
$t_{EHQZ}$	Data float after ENABLE	0	$48t_{CLCL}$	
$t_{GHGL}$	$\overline{\text{PROG}}$ high to $\overline{\text{PROG}}$ low	10		$\mu\text{s}$





## 80C51 Family Derivatives

## 8XC552/562 overview

### 8XC552 OVERVIEW

The 8XC552 is a stand-alone high-performance microcontroller designed for use in real-time applications such as instrumentation, industrial control, and automotive control applications such as engine management and transmission control. The device provides, in addition to the 80C51 standard functions, a number of dedicated hardware functions for these applications.

The 8XC552 single-chip 8-bit microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 8XC552 uses the powerful instruction set of the 80C51. Additional special function registers are incorporated to control the on-chip peripherals. Three versions of the derivative exist although the generic term "8XC552" is used to refer to family members:

83C552: 8k bytes mask-programmable ROM, 256 bytes RAM

87C552: 8k bytes EPROM, 256 bytes RAM

80C552: ROMless version of the 83C552

The 8XC552 contains a nonvolatile 8k × 8 read-only program memory, a volatile 256 × 8 read/write data memory, five 8-bit I/O ports and one 8-bit input port, two 16-bit timer/event counters (identical to the timers of the 80C51), an additional 16-bit timer coupled to capture and compare latches, a fifteen-source, two-priority-level, nested interrupt structure, an 8-input ADC, a dual DAC pulse width modulated interface, two serial interfaces (UART and I<sup>2</sup>C bus), a "watchdog" timer, and on-chip oscillator and timing circuits. For systems that require extra capability, the 8XC552 can be expanded using standard TTL compatible memories and logic.

The 8XC552 has two software selectable modes of reduced activity for further power reduction—Idle and Power-down. The Idle mode freezes the CPU and resets Timer T2 and the ADC and PWM circuitry but allows the other timers, RAM, serial ports, and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to become inoperative.

### 83C562 OVERVIEW

The 83C562 has been derived from the 8XC552 with the following changes:

- The SIO1 (I<sup>2</sup>C) interface has been omitted.
- The output of port lines P1.6 and P1.7 have a standard configuration instead of open drain.
- The resolution of the A/D converter is decreased from 10 bits to 8 bits.
- The time of an A/D conversion has decreased from 50 machine cycles to 24 machine cycles.

All other functions, pinning and packaging are unchanged.

This chapter of the users' guide can be used for the 83C562 by omitting or changing the following:

- Disregard the description of SIO1 (I<sup>2</sup>C).
  - The SFRs for the interface: S1ADR, S1DAT, S1STA, and S1CON are not implemented. The two SIO1 related flags ES1 in SFR IEN0 and PS1 in SFR IP0 are also not implemented. These two flag locations are undefined after RESET. The interrupt vector for SIO1 is not used.
  - Port lines P1.6 and P1.7 are not open drain but have the same standard configuration and electrical characteristics as P1.0-P1.5. Port lines P1.6 and P1.7 have alternative functions.
  - The A/D converter has a resolution of 8 bits instead of 10 bits and consequently the two high-order bits 6 and 7 of SFR ADCON are not implemented. These two locations are undefined after RESET. The 8-bit result of an A/D conversion is present in SFR ADCH. The result can always be calculated from the formula:
 
$$256 \times \frac{V_{IN} - AV_{ref-}}{AV_{ref+} - AV_{ref-}}$$
- The A/D conversion time is 24 machine cycles instead of 50 machine cycles, and the sampling time is 6 machine cycles instead of 8 machine cycles. The conversion time takes 3 machine cycles per bit.

- The serial I/O function SIO0 and its SFRs S0BUF and S0CON are renamed to SIO,

SBUF, and SCON. The interrupt related flags ES0 and PS0 are renamed ES and PS. Interrupt source S0 is renamed S. The serial I/O function remains the same.

### Differences From the 80C51

#### Program Memory

The 8XC552 contains 8k bytes of on-chip program memory which can be extended to 64k bytes with external memories (see Figure 1). When the EA pin is held high, the 8XC552 fetches instructions from internal ROM unless the address exceeds 1FFFFH. Locations 2000H to FFFFH are fetched from external program memory. When the EA pin is held low, all instruction fetches are from external memory. ROM locations 0003H to 0073H are used by interrupt service routines.

#### Data Memory

The internal data memory is divided into 3 sections: the lower 128 bytes of RAM, the upper 128 bytes of RAM, and the 128-byte special function register areas. The lower 128 bytes of RAM are directly and indirectly addressable. While RAM locations 128 to 255 and the special function register area share the same address space, they are accessed through different addressing modes. RAM locations 128 to 255 are only indirectly addressable, and the special function registers are only directly addressable. All other aspects of the internal RAM are identical to the 8051.

The stack may be located anywhere in the internal RAM by loading the 8-bit stack pointer. Stack depth is 256 bytes maximum.

#### Special Function Registers

The special function registers (directly addressable only) contain all of the 8XC552 registers except the program counter and the four register banks. Most of the 56 special function registers are used to control the on-chip peripheral hardware. Other registers include arithmetic registers (ACC, B, PSW), stack pointer (SP), and data pointer registers (DHP, DPL). Sixteen of the SFRs contain 128 directly addressable bit locations. Table 1 lists the 8XC552's special function registers.

The standard 80C51 SFRs are present and function identically in the 8XC552 except where noted in the following sections.

## 80C51 Family Derivatives

## 8XC552/562 overview

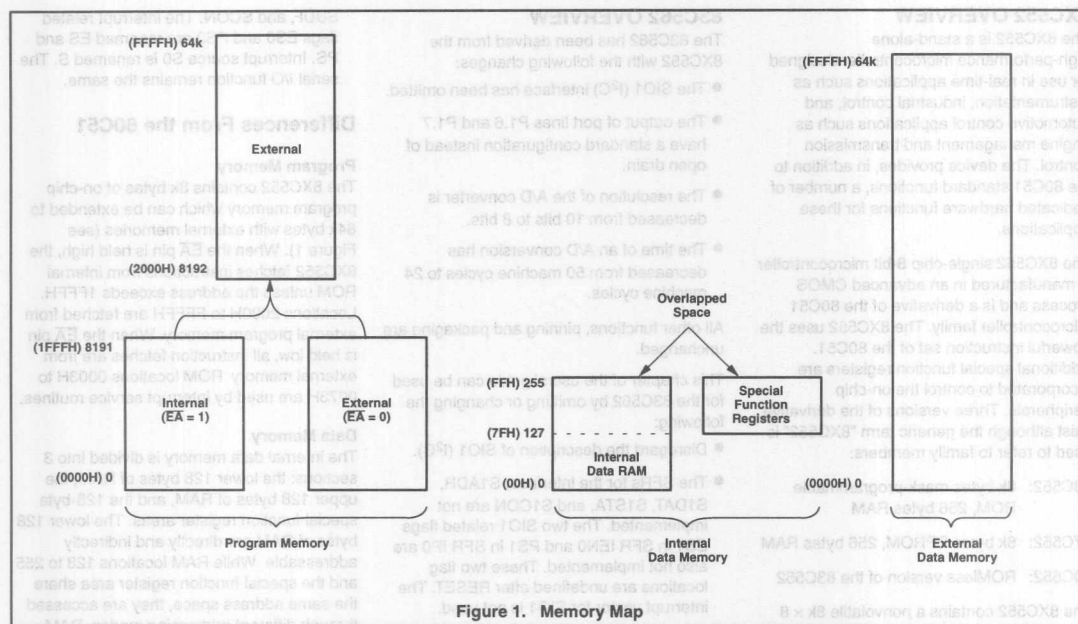


Figure 1. Memory Map

**Timer T2**

Timer T2 is a 16-bit timer consisting of two registers TMH2 (HIGH byte) and TML2 (LOW byte). The 16-bit timer/counter can be switched off or clocked via a prescaler from one of two sources:  $f_{osc}/12$  or an external signal. When Timer T2 is configured as a counter, the prescaler is clocked by an external signal on T2 (P1.4). A rising edge on T2 increments the prescaler, and the maximum repetition rate is one count per machine cycle (1MHz with a 12MHz oscillator).

The maximum repetition rate for Timer T2 is twice the maximum repetition rate for Timer 0 and Timer 1. T2 (P1.4) is sampled at S2P1 and again at S5P1 (i.e., twice per machine cycle). A rising edge is detected when T2 is LOW during one sample and HIGH during the next sample. To ensure that a rising edge is detected, the input signal must be LOW for at least 1/2 cycle and then HIGH for at least 1/2 cycle. If a rising edge is detected before the end of S2P1, the timer will be incremented during the following cycle; otherwise it will be incremented one cycle later. The prescaler

has a programmable division factor of 1, 2, 4, or 8 and is cleared if its division factor or input source is changed, or if the timer/counter is reset.

Timer T2 may be read "on the fly" but possesses no extra read latches, and software precautions may have to be taken to avoid misinterpretation in the event of an overflow from least to most significant byte while Timer T2 is being read. Timer T2 is not loadable and is reset by the RST signal or by a rising edge on the input signal RT2, if enabled. RT2 is enabled by setting bit T2ER (TM2CON.5).

When the least significant byte of the timer overflows or when a 16-bit overflow occurs, an interrupt request may be generated. Either or both of these overflows can be programmed to request an interrupt. In both cases, the interrupt vector will be the same. When the lower byte (TML2) overflows, flag T2B0 (TM2CON) is set and flag T2OV (TM2IR) is set when TMH2 overflows. These flags are set one cycle after an overflow occurs. Note that when T2OV is set, T2B0 will also be set. To enable the byte overflow

interrupt, bits ET2 (IE1.7, enable overflow interrupt, see Figure 2) and T2IS0 (TM2CON.6, byte overflow interrupt select) must be set. Bit TWB0 (TM2CON.4) is the Timer T2 byte overflow flag.

To enable the 16-bit overflow interrupt, bits ET2 (IE1.7, enable overflow interrupt) and T2IS1 (TM2CON.7, 16-bit overflow interrupt select) must be set. Bit T2OV (TM2IR.7) is the Timer T2 16-bit overflow flag. All interrupt flags must be reset by software. To enable both byte and 16-bit overflow, T2IS0 and T2IS1 must be set and two interrupt service routines are required. A test on the overflow flags indicates which routine must be executed. For each routine, only the corresponding overflow flag must be cleared.

Timer T2 may be reset by a rising edge on RT2 (P1.5) if the Timer T2 external reset enable bit (T2ER) in T2CON is set. This reset also clears the prescaler. In the idle mode, the timer/counter and prescaler are reset and halted. Timer T2 is controlled by the TM2CON special function register (see Figure 3).

## 80C51 Family Derivatives

## 8XC552/562 overview

Table 1. 8XC552 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB							LSB	
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
ADCH#	A/D converter high	C6H									xxxxxxxB
ADCON#	Adc control	C5H	ADC.1	ADC.0	ADEX	ADCI	ADCS	AADR2	AADR1	AADR0	xx000000B
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
CTCON#	Capture control	EBH	CTN3	CTP3	CTN2	CTP2	CTN1	CTP1	CTN0	CTP0	00H
CTH3#	Capture high 3	CFH									xxxxxxxB
CTH2#	Capture high 2	CEH									xxxxxxxB
CTH1#	Capture high 1	CDH									xxxxxxxB
CTH0#	Capture high 0	CCH									xxxxxxxB
CMH2#	Compare high 2	CBH									00H
CMH1#	Compare high 1	CAH									00H
CMH0#	Compare high 0	C9H									00H
CTL3#	Capture low 3	AFH									xxxxxxxB
CTL2#	Capture low 2	AEH									xxxxxxxB
CTL1#	Capture low 1	ADH									xxxxxxxB
CTL0#	Capture low 0	ACH									xxxxxxxB
CML2#	Compare low 2	ABH									00H
CML1#	Compare low 1	AAH									00H
CML0#	Compare low 0	A9H									00H
DPTR:	Data pointer (2 bytes)										
DPH	Data pointer high	83H									00H
DPL	Data pointer low	82H									00H
			AF	AE	AD	AC	AB	AA	A9	A8	
IEN0*#	Interrupt enable 0	A8H	EA	EAD	ES1	ES0	ET1	EX1	ET0	EX0	00H
			EF	EE	ED	EC	EB	EA	E9	E8	
IEN1*#	Interrupt enable 1	E8H	ET2	ECM2	ECM1	ECM0	ECT3	ECT2	ECT1	ECT0	00H
			BF	BE	BD	BC	BB	BA	B9	B8	
IP0*#	Interrupt priority 0	B8H	-	PAD	PS1	PS0	PT1	PX1	PT0	PX0	x0000000B
			FF	FE	FD	FC	FB	FA	F9	F8	
IP1*#	Interrupt priority 1	F8H	PT2	PCM2	PCM1	PCM0	PCT3	PCT2	PCT1	PCT0	00H
P5#	Port 5	C4H	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	xxxxxxxB
			C7	C6	C5	C4	C3	C2	C1	C0	
P4#	Port 4	C0H	CMT1	CMT0	CMSR5	CMSR4	CMSR3	CMSR2	CMSR1	CMSR0	FFH
			B7	B6	B5	B4	B3	B2	B1	B0	
P3*	Port 3	B0H	RD	WR	T1	T0	INT1	INT0	TXD	RXD	FFH
			A7	A6	A5	A4	A3	A2	A1	A0	
P2*	Port 2	A0H	A15	A14	A13	A12	A11	A10	A9	A8	FFH
			97	96	95	94	93	92	91	90	
P1*	Port 1	90H	SDA	SCL	RT2	T2	CT3I	CT2I	CT1I	CT0I	FFH
			87	86	85	84	83	82	81	80	
P0*	Port 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
PCON#	Power control	87H	SMOD	-	-	WLE	GF1	GF0	PD	IDL	00xx0000B
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program status word	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	00H

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.

80C51 Family Derivatives

8XC552/562 overview

Table 1. 8XC552 Special Function Registers (Continued)

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE	
			MSB				LSB					
PWMP#	PWM prescaler	FEH									00H	
PWM1#	PWM register 1	FDH									00H	
PWM0#	PWM register 0	FCH									00H	
RTE#	Reset/toggle enable	EFH	TP47	TP46	RP45	RP44	RP43	RP42	RP41	RP40	00H	
SP	Stack pointer	81H									07H	
S0BUF	Serial 0 data buffer	99H									xxxxxxxB	
			9F	9E	9D	9C	9B	9A	99	98		
S0CON*	Serial 0 control	98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00H	
S1ADR#	Serial 1 address	DBH	SLAVE ADDRESS								GC	00H
SIDAT#	Serial 1 data	DAH										00H
S1STA#	Serial 1 status	D9H	SC4	SC3	SC2	SC1	SC0	0	0	0	F8H	
			DF	DE	DD	DC	DB	DA	D9	D8		
SICON#*	Serial 1 control	D8H	CR2	ENS1	STA	ST0	SI	AA	CR1	CR0	00H	
STE#	Set enable	EEH	TG47	TG46	SP45	SP44	SP43	SP42	SP41	SP40	C0H	
TH1	Timer high 1	8DH									00H	
TH0	Timer high 0	8CH									00H	
TL1	Timer low 1	8BH									00H	
TL0	Timer low 0	8AH									00H	
TMH2#	Timer high 2	EDH									00H	
TML2#	Timer low 2	ECH									00H	
TMOD	Timer mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H	
			8F	8E	8D	8C	8B	8A	89	88		
TCON*	Timer control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H	
TM2CON#	Timer 2 control	EAH	T2IS1	T2IS0	T2ER	T2B0	T2P1	T2P0	T2MS1	T2MS0	00H	
			CF	CE	CD	CC	CB	CA	C9	C8		
TM2IR#*	Timer 2 int flag reg	C8H	T20V	CM12	CM11	CM10	CT13	CT12	CT11	CT10	00H	
T3#	Timer 3	FFH										00H

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.

7 6 5 4 3 2 1 0									
IEN1 (E8H)		ET2	ECM2	ECM1	ECM0	ECT3	ECT2	ECT1	ECT0
		(MSB)				(LSB)			
		Bit	Symbol		Function				
		IEN1.7	ET2		Enable Timer T2 overflow interrupt(s)				
		IEN1.6	ECM2		Enable T2 Comparator 2 interrupt				
		IEN1.5	ECM1		Enable T2 Comparator 1 interrupt				
		IEN1.4	ECM0		Enable T2 Comparator 0 interrupt				
		IEN1.3	ECT3		Enable T2 Capture register 3 interrupt				
		IEN1.2	ECT2		Enable T2 Capture register 2 interrupt				
		IEN1.1	ECT1		Enable T2 Capture register 1 interrupt				
		IEN1.0	ECT0		Enable T2 Capture register 0 interrupt				

Figure 2. Timer T2 Interrupt Enable Register (IEN1)

Figure 2. Timer T2 Interrupt Enable Register (IEN1)

## 80C51 Family Derivatives

## 8XC552/562 overview

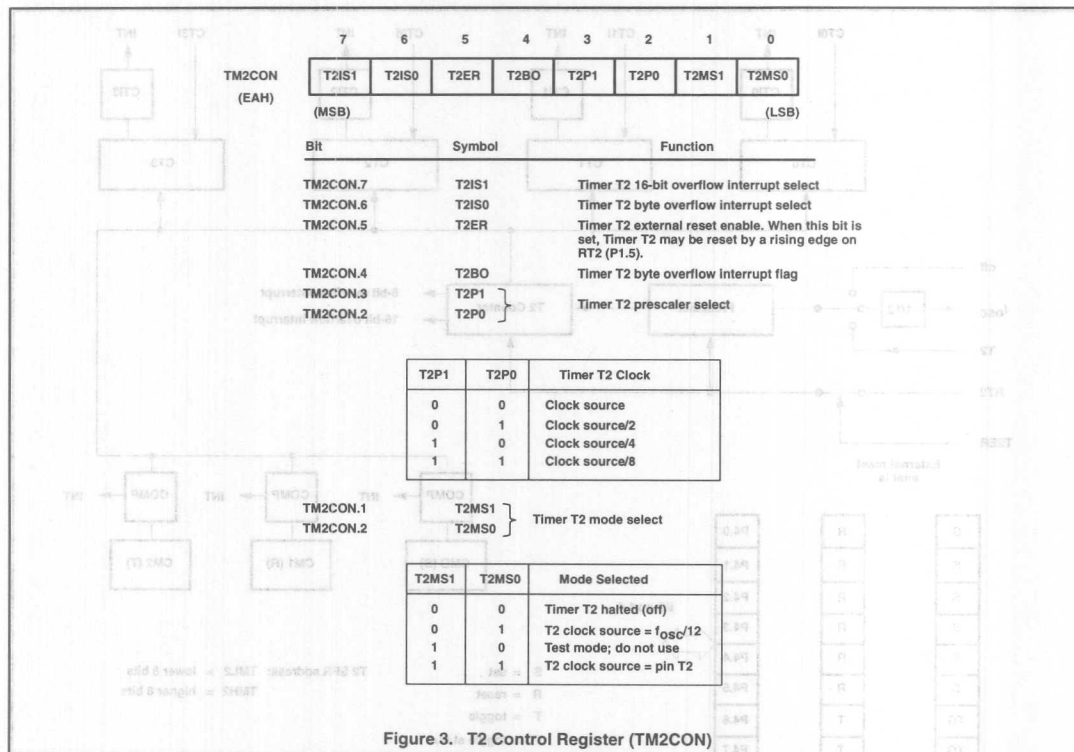


Figure 3. T2 Control Register (TM2CON)

**Timer T2 Extension:** When a 12MHz oscillator is used, a 16-bit overflow on Timer T2 occurs every 65.5, 131, 262, or 524 ms, depending on the prescaler division ratio; i.e., the maximum cycle time is approximately 0.5 seconds. In applications where cycle times are greater than 0.5 seconds, it is necessary to extend Timer T2. This is achieved by selecting  $f_{osc}/12$  as the clock source (set T2MS0, reset T2MS1), setting the prescaler division ratio to 1/8 (set T2P0, set T2P1), disabling the byte overflow interrupt (reset T2IS0) and enabling the 16-bit overflow interrupt (set T2IS1). The following software routine is written for a three-byte extension which gives a maximum cycle time of approximately 2400 hours.

```
OVINT: PUSH ACC ;save accumulator
      PUSH PSW ;save status
      INC A ;increment first
      MOV A, TIMEX1 ;byte (low order)
      MOV A, TIMEX1 ;of extended timer
      JNZ INTEX ;jump to INTEX if
      ;there is no
      ;overflow
```

```
INC TIMEX2 ;increment second
      ;byte
MOV A, TIMEX2
JNZ INTEX ;jump to INTEX if
      ;there is no
      ;overflow
INC TIMEX3 ;increment third
      ;byte (high order)

INTEX: CLR T2OV ;reset interrupt
      ;flag
POP PSW ;restore status
POP ACC ;restore accumulator
      RETI ;return from
      ;interrupt
```

**Timer T2, Capture and Compare Logic:**

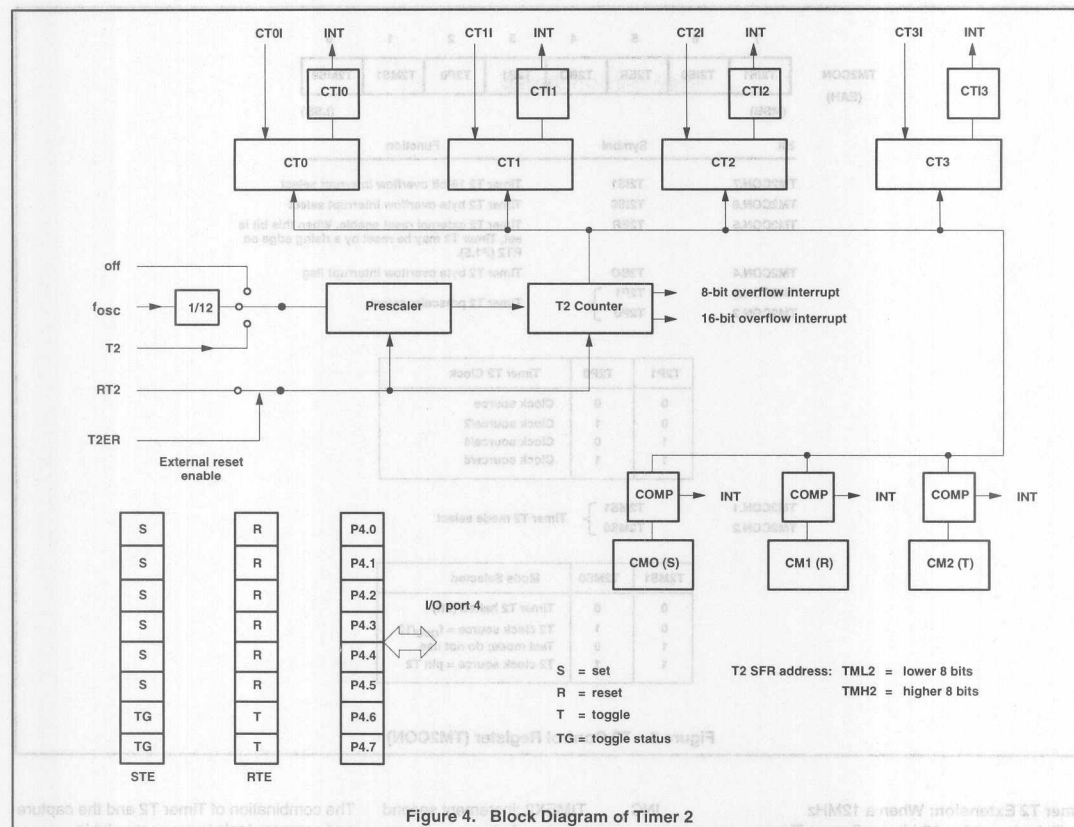
Timer T2 is connected to four 16-bit capture registers and three 16-bit compare registers. A capture register may be used to capture the contents of Timer T2 when a transition occurs on its corresponding input pin. A compare register may be used to set, reset, or toggle port 4 output pins at certain pre-programmable time intervals.

The combination of Timer T2 and the capture and compare logic is very powerful in applications involving rotating machinery, automotive injection systems, etc. Timer T2 and the capture and compare logic are shown in Figure 4.

**Capture Logic:** The four 16-bit capture registers that Timer T2 is connected to are: CT0, CT1, CT2, and CT3. These registers are loaded with the contents of Timer T2, and an interrupt is requested upon receipt of the input signals CT0I, CT1I, CT2I, or CT3I. These input signals are shared with port 1. The four interrupt flags are in the Timer T2 interrupt register (TM2IR special function register). If the capture facility is not required, these inputs can be regarded as additional external interrupt inputs.

Using the capture control register CTCON (see Figure 5), these inputs may capture on a rising edge, a falling edge, or on either a rising or falling edge. The inputs are sampled during S1P1 of each cycle. When a selected edge is detected, the contents of Timer T2 are captured at the end of the cycle.





**Measuring Time Intervals Using Capture Registers:** When a recurring external event is represented in the form of rising or falling edges on one of the four capture pins, the time between two events can be measured using Timer T2 and a capture register. When an event occurs, the contents of Timer T2 are copied into the relevant capture register and an interrupt request is generated. The interrupt service routine may then compute the interval time if it knows the previous contents of Timer T2 when the last event occurred. With a 12MHz oscillator, Timer T2 can be programmed to overflow every 524ms. When event interval times are shorter than this, computing the interval time is simple, and the interrupt service routine is short. For longer interval times, the Timer T2 extension routine may be used.

**Compare Logic:** Each time Timer T2 is incremented, the contents of the three 16-bit compare registers CM0, CM1, and CM2 are compared with the new counter value of

Timer T2. When a match is found, the corresponding interrupt flag in TM2IR is set at the end of the following cycle. When a match with CM0 occurs, the controller sets bits 0-5 of port 4 if the corresponding bits of the set enable register STE are at logic 1.

When a match with CM1 occurs, the controller resets bits 0-5 of port 4 if the corresponding bits of the reset/toggle enable register RTE are at logic 1 (see Figure 6 for RTE register function). If RTE is "0", then P4.n is not affected by a match between CM1 or CM2 and Timer 2. When a match with CM2 occurs, the controller "toggles" bits 6 and 7 of port 4 if the corresponding bits of the RTE are at logic 1. The port latches of bits 6 and 7 are not toggled. Two additional flip-flops store the last operation, and it is these flip-flops that are toggled.

Thus, if the current operation is "set," the next operation will be "reset" even if the port latch is reset by software before the "reset" operation occurs. The first "toggle" after a

chip RESET will set the port latch. The contents of these two flip-flops can be read at STE.6 and STE.7 (corresponding to P4.6 and P4.7, respectively). Bits STE.6 and STE.7 are read only (see Figure 7 for STE register function). A logic 1 indicates that the next toggle will set the port latch; a logic 0 indicates that the next toggle will reset the port latch. CM0, CM1, and CM2 are reset by the RST signal.

The modified port latch information appears at the port pin during S5P1 of the cycle following the cycle in which a match occurred. If the port is modified by software, the outputs change during S1P1 of the following cycle. Each port 4 bit can be set or reset by software at any time. A hardware modification resulting from a comparator match takes precedence over a software modification in the same cycle. When the comparator results require a "set" and a "reset" at the same time, the port latch will be reset.

## 80C51 Family Derivatives

## 8XC552/562 overview

**Timer T2 Interrupt Flag Register TM2IR:**

Eight of the nine Timer T2 interrupt flags are located in special function register TM2IR (see Figure 8). The ninth flag is TM2CON.4.

The CT0I and CT1I flags are set during S4 of the cycle in which the contents of Timer T2 are captured. CT0I is scanned by the interrupt logic during S2, and CT1I is scanned during S3. CT2I and CT3I are set during S6 and are scanned during S4 and S5. The associated interrupt requests are recognized during the following cycle. If these

flags are polled, a transition at CT0I or CT1I will be recognized one cycle before a transition on CT2I or CT3I since registers are read during S5. The CMI0, CMI1, and CMI2 flags are set during S6 of the cycle following a match. CMI0 is scanned by the interrupt logic during S2; CMI1 and CMI2 are scanned during S3 and S4. A match will be recognized by the interrupt logic (or by polling the flags) two cycles after the match takes place.

The 16-bit overflow flag (T2OV) and the byte

overflow flag (T2BO) are set during S6 of the cycle in which the overflow occurs. These flags are recognized by the interrupt logic during the next cycle.

Special function register IP1 (Figure 8) is used to determine the Timer T2 interrupt priority. Setting a bit high gives that function a high priority, and setting a bit low gives the function a low priority. The functions controlled by the various bits of the IP1 register are shown in Figure 8.

7 6 5 4 3 2 1 0							
CTCON (E8H)							
CTN3	CTP3	CTN2	CTP2	CTN1	CTP1	CTN0	CTP0
(MSB)				(LSB)			
Bit	Symbol	Capture/Interrupt on:					
CTCON.7	CTN3	Capture Register 3 triggered by a falling edge on CT3I					
CTCON.6	CTP3	Capture Register 3 triggered by a rising edge on CT3I					
CTCON.5	CTN2	Capture Register 2 triggered by a falling edge on CT2I					
CTCON.4	CTP2	Capture Register 2 triggered by a rising edge on CT2I					
CTCON.3	CTN1	Capture Register 1 triggered by a falling edge on CT1I					
CTCON.2	CTP1	Capture Register 1 triggered by a rising edge on CT1I					
CTCON.1	CTN0	Capture Register 0 triggered by a falling edge on CT0I					
CTCON.0	CTP0	Capture Register 0 triggered by a rising edge on CT0I					

**Figure 5. Capture Control Register (CTCON)**

7 6 5 4 3 2 1 0							
RTE (EFH)							
TP47	TP46	RP45	RP44	RP43	RP42	RO41	RP40
(MSB)				(LSB)			
Bit	Symbol	Function					
RTE.7	TP47	If "1" then P4.7 toggles on a match between CM2 and Timer T2					
RTE.6	TP46	If "1" then P4.6 toggles on a match between CM2 and Timer T2					
RTE.5	RP45	If "1" then P4.5 is reset on a match between CM1 and Timer T2					
RTE.4	RP44	If "1" then P4.4 is reset on a match between CM1 and Timer T2					
RTE.3	RP43	If "1" then P4.3 is reset on a match between CM1 and Timer T2					
RTE.2	RP42	If "1" then P4.2 is reset on a match between CM1 and Timer T2					
RTE.1	RP41	If "1" then P4.1 is reset on a match between CM1 and Timer T2					
RTE.0	RP40	If "1" then P4.0 is reset on a match between CM1 and Timer T2					

**Figure 6. Reset/Toggle Enable Register (RTE)**

7 6 5 4 3 2 1 0							
STE (EEH)							
TG47 TG46 SP45 SP44 SP43 SP42 SP41 SP40							
(MSB)				(LSB)			
Bit	Symbol	Function					
STE.7	TG47	Toggle flip-flops					
STE.6	TG46	Toggle flip-flops					
STE.5	SP45	If "1" then P4.5 is set on a match between CM0 and Timer T2					
STE.4	SP44	If "1" then P4.4 is set on a match between CM0 and Timer T2					
STE.3	SP43	If "1" then P4.3 is set on a match between CM0 and Timer T2					
STE.2	SP42	If "1" then P4.2 is set on a match between CM0 and Timer T2					
STE.1	SP41	If "1" then P4.1 is set on a match between CM0 and Timer T2					
STE.0	SP40	If "1" then P4.0 is set on a match between CM0 and Timer T2					

**Figure 7. Set Enable Register (STE)**

## 80C51 Family Derivatives

## 8XC552/562 overview

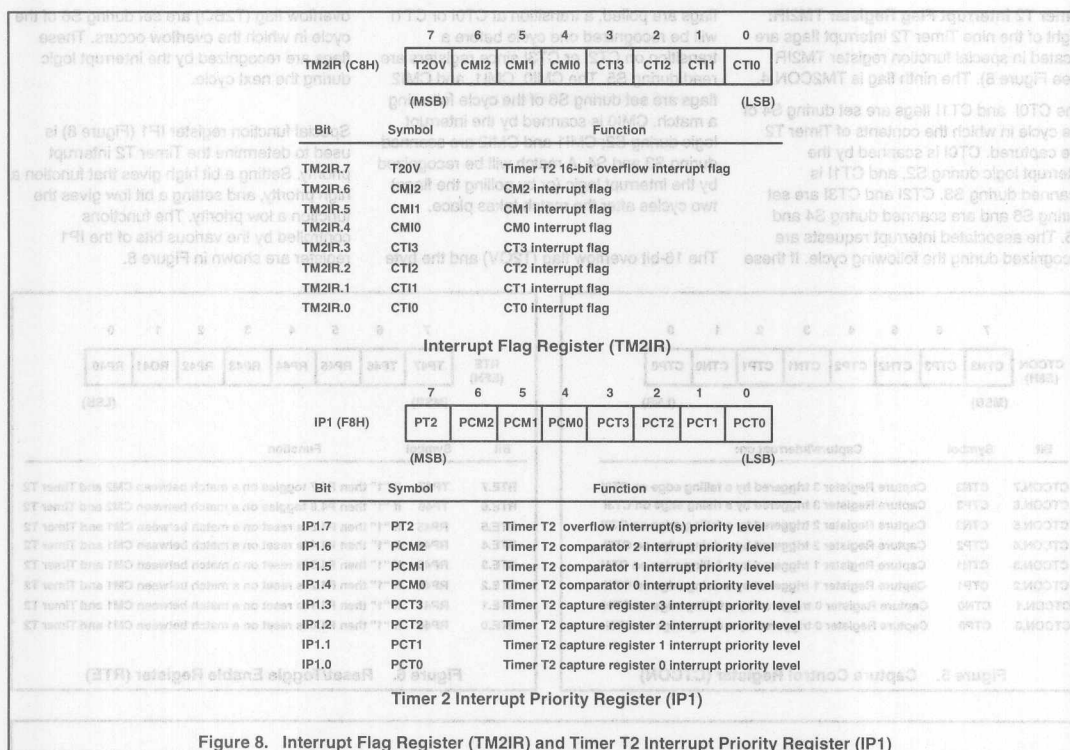


Figure 8. Interrupt Flag Register (TM2IR) and Timer T2 Interrupt Priority Register (IP1)

## Timer T3, The Watchdog Timer

In addition to Timer T2 and the standard timers, a watchdog timer is also incorporated on the 8XC552. The purpose of a watchdog timer is to reset the microcontroller if it enters erroneous processor states (possibly caused by electrical noise or RFI) within a reasonable period of time. An analogy is the "dead man's handle" in railway locomotives. When enabled, the watchdog circuitry will generate a system reset if the user program fails to reload the watchdog timer within a specified length of time known as the "watchdog interval."

**Watchdog Circuit Description:** The watchdog timer (Timer T3) consists of an 8-bit timer with an 11-bit prescaler as shown in Figure 9. The prescaler is fed with a signal whose frequency is 1/12 the oscillator frequency (1MHz with a 12MHz oscillator). The 8-bit timer is incremented every "t" seconds, where:

$$t = 12 \times 2048 \times 1/f_{OSC}$$

$$(\approx 1.5\text{ms at } f_{OSC} = 16\text{MHz};$$

$$= 1\text{ms at } f_{OSC} = 24\text{MHz})$$

If the 8-bit timer overflows, a short internal reset pulse is generated which will reset the 8XC552. A short output reset pulse is also generated at the RST pin. This short output pulse (3 machine cycles) may be destroyed if the RST pin is connected to a capacitor. This would not, however, affect the internal reset operation.

Watchdog operation is activated when external pin EW is tied low. When EW is tied low, it is impossible to disable the watchdog operation by software.

**How to Operate the Watchdog Timer:** The watchdog timer has to be reloaded within periods that are shorter than the programmed watchdog interval; otherwise the watchdog timer will overflow and a system reset will be generated. The user program must therefore continually execute sections of code which reload the watchdog timer. The period of time elapsed between execution of these sections of code must never exceed the watchdog interval. When using a 16MHz oscillator, the watchdog interval is programmable between 1.5ms and 392ms. When using a 24MHz oscillator, the watchdog interval is

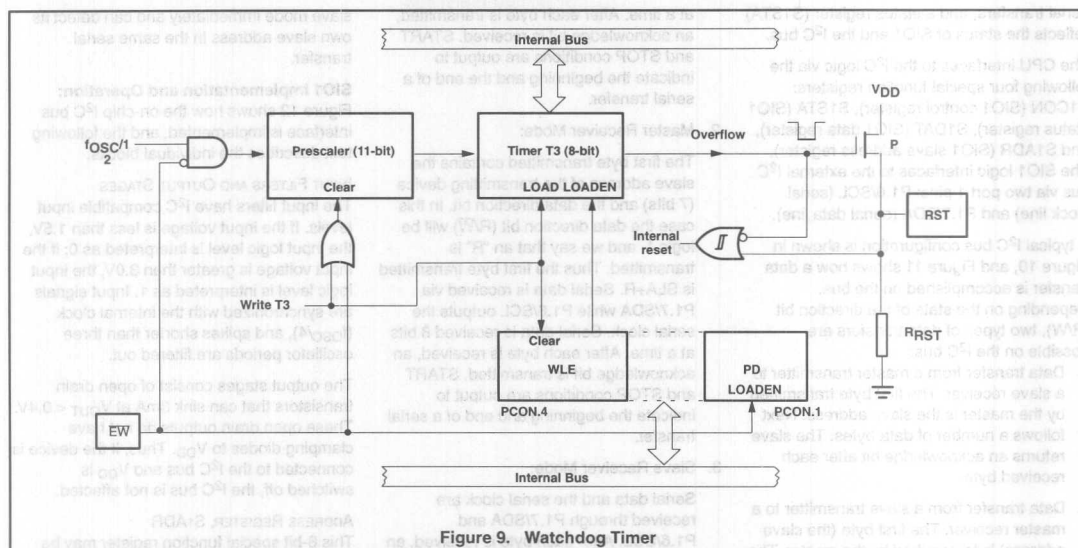
programmable between 1ms and 255ms.

In order to prepare software for watchdog operation, a programmer should first determine how long his system can sustain an erroneous processor state. The result will be the maximum watchdog interval. As the maximum watchdog interval becomes shorter, it becomes more difficult for the programmer to ensure that the user program always reloads the watchdog timer within the watchdog interval, and thus it becomes more difficult to implement watchdog operation.

The programmer must now partition the software in such a way that reloading of the watchdog is carried out in accordance with the above requirements. The programmer must determine the execution times of all software modules. The effect of possible conditional branches, subroutines, external and internal interrupts must all be taken into account. Since it may be very difficult to evaluate the execution times of some sections of code, the programmer should use worst case estimations. In any event, the programmer must make sure that the watchdog is not activated during normal operation.

## 80C51 Family Derivatives

## 8XC552/562 overview



### Figure 9. Watchdog Time

The watchdog timer is reloaded in two stages in order to prevent erroneous software from reloading the watchdog. First PCON.4 (WLE) must be set. The T3 may be loaded. When T3 is loaded, PCON.4 (WLE) is automatically reset. T3 cannot be loaded if PCON.4 (WLE) is reset. Reload code may be put in a subroutine as it is called frequently. Since Timer T3 is an up-counter, a reload value of 00H gives the maximum watchdog interval (510ms with a 12MHz oscillator), and a reload value of 0FFH gives the minimum watchdog interval (2ms with a 12MHz oscillator).

In the idle mode, the watchdog circuitry remains active. When watchdog operation is implemented, the power-down mode cannot be used since both states are contradictory. Thus, when watchdog operation is enabled by tying external pin  $\overline{EW}$  low, it is impossible to enter the power-down mode, and an attempt to set the power-down bit (PCON.1) will have no effect. PCON.1 will remain at logic 0.

During the early stages of software development/debugging, the watchdog may be disabled by tying the  $\overline{\text{EW}}$  pin high. At a later stage,  $\overline{\text{EW}}$  may be tied low to complete the debugging process.

**Watchdog Software Example:** The following example shows how watchdog operation might be handled in a user program.

```

;at the program start:
T3      EQU 0FFH ;address of
          ;watchdog
          ;timer T3
PCON     EQU 087H ;address of
          ;PCON SFR
WATCH-INTV EQU 156 ;watchdog
          ;interval
          ;(e.g., 2x100ms)

```

;to be inserted at each watchdog reload  
;location within the user program:

## LCALL WATCHDOG

```
:watchdog service routine:
```

```
WATCHDOG: ORL PCON,#10H ;set
                                ;condition
                                ;flag
                                ;(PCON.4)
MOV T3,WATCH-INV ;load T3
                                ;with
                                ;watchdog
                                ;interval
RET
```

If it is possible for this subroutine to be called in an erroneous state, then the condition flag WLE should be set at different parts of the main program.

## Serial I/O

The 8XC552 is equipped with two independent serial ports: SIO0 and SIO1. SIO0 is a full duplex UART port and is

identical to the 80C51 serial port. SIO1 accommodates the I<sup>2</sup>C bus.

**SIO0:** SIO0 is a full duplex serial I/O port identical to that on the 80C51. Its operation is the same, including the use of timer 1 as a baud rate generator.

**SIO1, I2C Serial I/O:** The I<sup>2</sup>C bus uses two wires (SDA and SCL) to transfer information between devices connected to the bus. The main features of the bus are:

- Bidirectional data transfer between masters and slaves
- Multimaster bus (no central master)
- Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
- Serial clock synchronization allows devices with different bit rates to communicate via one serial bus
- Serial clock synchronization can be used as a handshake mechanism to suspend and resume serial transfer
- The I<sup>2</sup>C bus may be used for test and diagnostic purposes

The output latches of P1.6 and P1.7 must be set to logic 1 in order to enable SIO1.

The 8XC552 on-chip <sup>12</sup>C logic provides a serial interface that meets the <sup>12</sup>C bus specification and supports all transfer modes (other than the low-speed mode) from and to the <sup>12</sup>C bus. The SIO1 logic handles bytes transfer autonomously. It also keeps track of



## 80C51 Family Derivatives

## 8XC552/562 overview

serial transfers, and a status register (S1STA) reflects the status of SIO1 and the I<sup>2</sup>C bus.

The CPU interfaces to the I<sup>2</sup>C logic via the following four special function registers: S1CON (SIO1 control register), S1STA (SIO1 status register), S1DAT (SIO1 data register), and S1ADR (SIO1 slave address register). The SIO1 logic interfaces to the external I<sup>2</sup>C bus via two port 1 pins: P1.6/SCL (serial clock line) and P1.7/SDA (serial data line).

A typical I<sup>2</sup>C bus configuration is shown in Figure 10, and Figure 11 shows how a data transfer is accomplished on the bus.

Depending on the state of the direction bit (R/W), two types of data transfers are possible on the I<sup>2</sup>C bus:

1. Data transfer from a master transmitter to a slave receiver. The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte.
2. Data transfer from a slave transmitter to a master receiver. The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. Next follows the data bytes transmitted by the slave to the master. The master returns an acknowledge bit after all received bytes other than the last byte. At the end of the last received byte, a "not acknowledge" is returned.

The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the I<sup>2</sup>C bus will not be released.

**Modes of Operation:** The on-chip SIO1 logic may operate in the following four modes:

#### 1. Master Transmitter Mode:

Serial data output through P1.7/SDA while P1.6/SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the data direction bit. In this case the data direction bit (R/W) will be logic 0, and we say that a "W" is transmitted. Thus the first byte transmitted is SLA+W. Serial data is transmitted 8 bits

at a time. After each byte is transmitted, an acknowledge bit is received. START and STOP conditions are output to indicate the beginning and the end of a serial transfer.

#### 2. Master Receiver Mode:

The first byte transmitted contains the slave address of the transmitting device (7 bits) and the data direction bit. In this case the data direction bit (R/W) will be logic 1, and we say that an "R" is transmitted. Thus the first byte transmitted is SLA+R. Serial data is received via P1.7/SDA while P1.6/SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions are output to indicate the beginning and end of a serial transfer.

#### 3. Slave Receiver Mode:

Serial data and the serial clock are received through P1.7/SDA and P1.6/SCL. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and direction bit.

#### 4. Slave Transmitter Mode:

The first byte is received and handled as in the slave receiver mode. However, in this mode, the direction bit will indicate that the transfer direction is reversed. Serial data is transmitted via P1.7/SDA while the serial clock is input through P1.6/SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer.

In a given application, SIO1 may operate as a master and as a slave. In the slave mode, the SIO1 hardware looks for its own slave address and the general call address. If one of these addresses is detected, an interrupt is requested. When the microcontroller wishes to become the bus master, the hardware waits until the bus is free before the master mode is entered so that a possible slave action is not interrupted. If bus arbitration is lost in the master mode, SIO1 switches to the

slave mode immediately and can detect its own slave address in the same serial transfer.

#### SIO1 Implementation and Operation:

Figure 12 shows how the on-chip I<sup>2</sup>C bus interface is implemented, and the following text describes the individual blocks.

#### INPUT FILTERS AND OUTPUT STAGES

The input filters have I<sup>2</sup>C compatible input levels. If the input voltage is less than 1.5V, the input logic level is interpreted as 0; if the input voltage is greater than 3.0V, the input logic level is interpreted as 1. Input signals are synchronized with the internal clock ( $f_{OSC}/4$ ), and spikes shorter than three oscillator periods are filtered out.

The output stages consist of open drain transistors that can sink 3mA at  $V_{OUT} < 0.4V$ . These open drain outputs do not have clamping diodes to  $V_{DD}$ . Thus, if the device is connected to the I<sup>2</sup>C bus and  $V_{DD}$  is switched off, the I<sup>2</sup>C bus is not affected.

#### ADDRESS REGISTER, S1ADR

This 8-bit special function register may be loaded with the 7-bit slave address (7 most significant bits) to which SIO1 will respond when programmed as a slave transmitter or receiver. The LSB (GC) is used to enable general call address (00H) recognition.

#### COMPARATOR

The comparator compares the received 7-bit slave address with its own slave address (7 most significant bits in S1ADR). It also compares the first received 8-bit byte with the general call address (00H). If an equality is found, the appropriate status bits are set and an interrupt is requested.

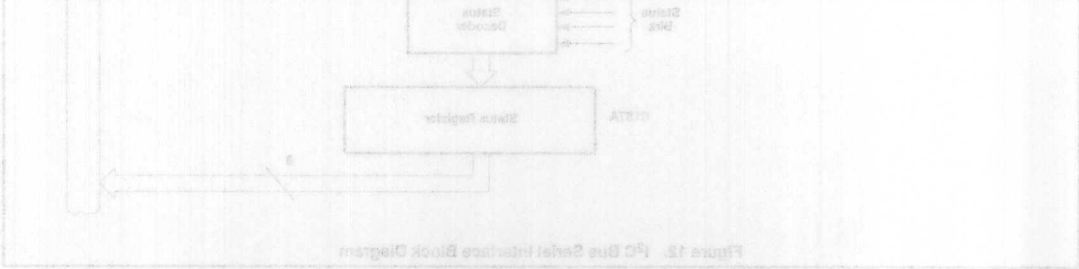
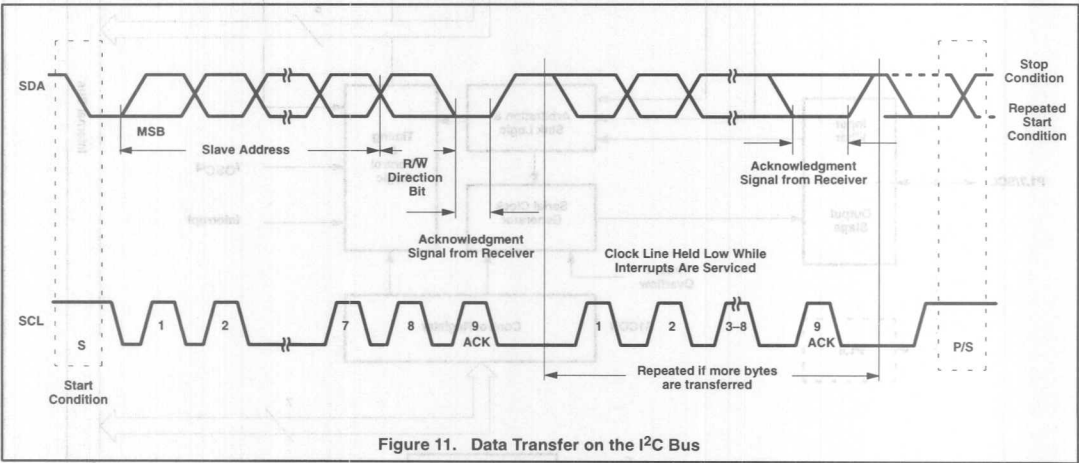
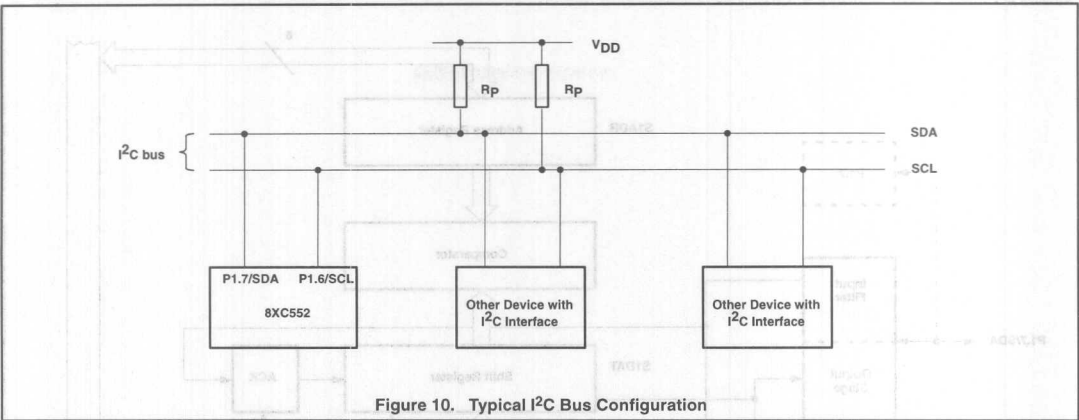
#### SHIFT REGISTER, S1DAT

This 8-bit special function register contains a byte of serial data to be transmitted or a byte which has just been received. Data in S1DAT is always shifted from right to left; the first bit to be transmitted is the MSB (bit 7) and, after a byte has been received, the first bit of received data is located at the MSB of S1DAT. While data is being shifted out, data on the bus is simultaneously being shifted in; S1DAT always contains the last byte present on the bus. Thus, in the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data in S1DAT.



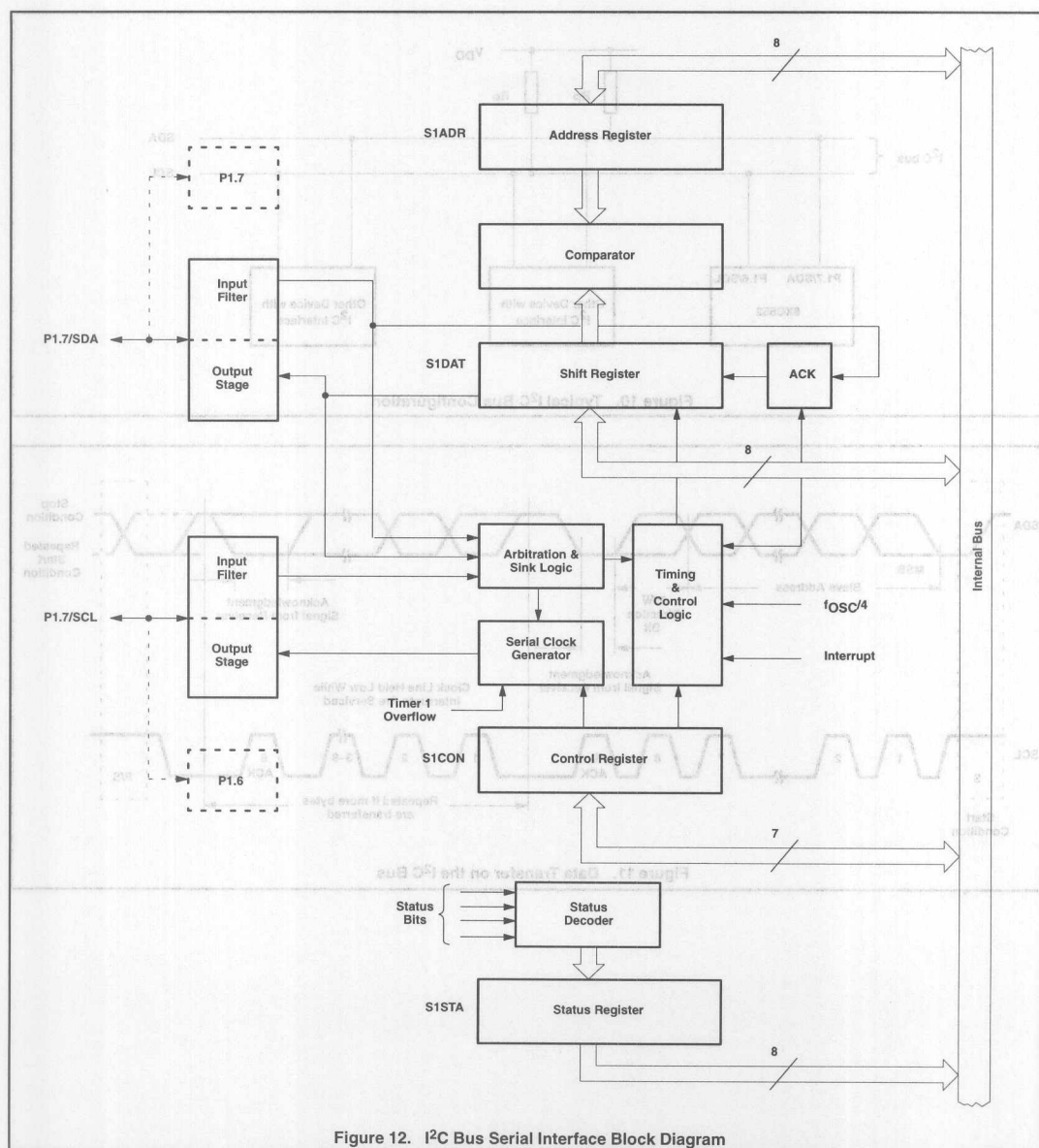
80C51 Family Derivatives

89V 8XC552/562 overview

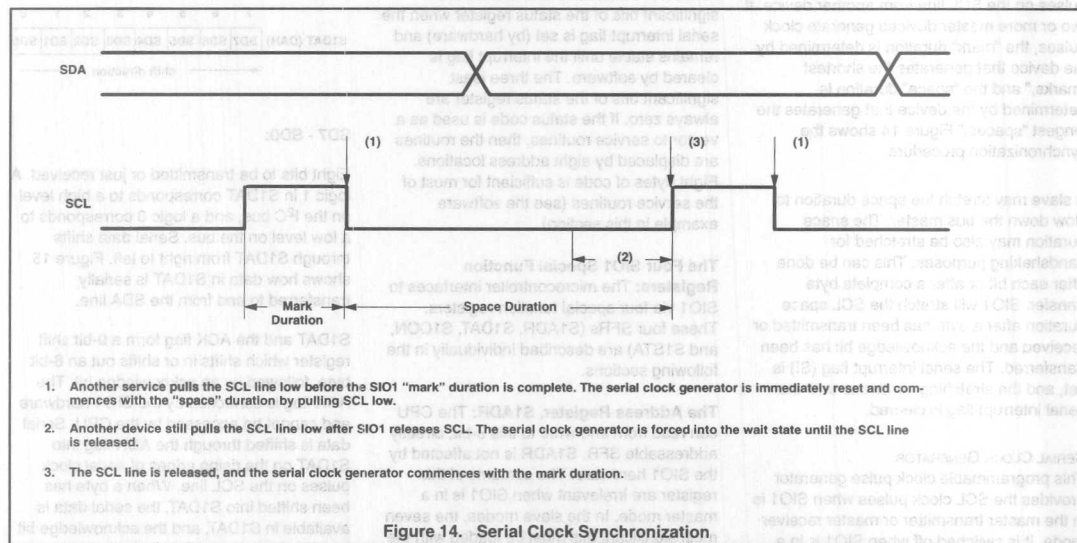
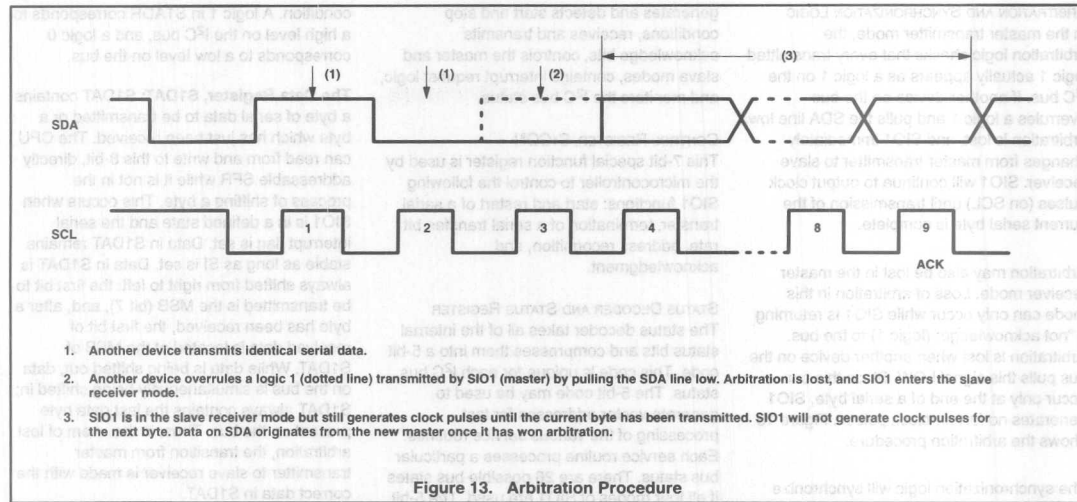


## 80C51 Family Derivatives

## 8XC552/562 overview



**Figure 12. I<sup>2</sup>C Bus Serial Interface Block Diagram**



## 80C51 Family Derivatives

## 8XC552/562 overview

### ARBITRATION AND SYNCHRONIZATION LOGIC

In the master transmitter mode, the arbitration logic checks that every transmitted logic 1 actually appears as a logic 1 on the I<sup>2</sup>C bus. If another device on the bus overrules a logic 1 and pulls the SDA line low, arbitration is lost, and SIO1 immediately changes from master transmitter to slave receiver. SIO1 will continue to output clock pulses (on SCL) until transmission of the current serial byte is complete.

Arbitration may also be lost in the master receiver mode. Loss of arbitration in this mode can only occur while SIO1 is returning a "not acknowledge: (logic 1) to the bus. Arbitration is lost when another device on the bus pulls this signal LOW. Since this can occur only at the end of a serial byte, SIO1 generates no further clock pulses. Figure 13 shows the arbitration procedure.

The synchronization logic will synchronize the serial clock generator with the clock pulses on the SCL line from another device. If two or more master devices generate clock pulses, the "mark" duration is determined by the device that generates the shortest "marks," and the "space" duration is determined by the device that generates the longest "spaces." Figure 14 shows the synchronization procedure.

A slave may stretch the space duration to slow down the bus master. The space duration may also be stretched for handshaking purposes. This can be done after each bit or after a complete byte transfer. SIO1 will stretch the SCL space duration after a byte has been transmitted or received and the acknowledge bit has been transferred. The serial interrupt flag (SI) is set, and the stretching continues until the serial interrupt flag is cleared.

### SERIAL CLOCK GENERATOR

This programmable clock pulse generator provides the SCL clock pulses when SIO1 is in the master transmitter or master receiver mode. It is switched off when SIO1 is in a slave mode. The programmable output clock frequencies are:  $f_{OSC}/120$ ,  $f_{OSC}/9600$ , and the Timer 1 overflow rate divided by eight. The output clock pulses have a 50% duty cycle unless the clock generator is synchronized with other SCL clock sources as described above.

### TIMING AND CONTROL

The timing and control logic generates the timing and control signals for serial byte handling. This logic block provides the shift pulses for S1DAT, enables the comparator,

generates and detects start and stop conditions, receives and transmits acknowledge bits, controls the master and slave modes, contains interrupt request logic, and monitors the I<sup>2</sup>C bus status.

### CONTROL REGISTER, S1CON

This 7-bit special function register is used by the microcontroller to control the following SIO1 functions: start and restart of a serial transfer, termination of a serial transfer, bit rate, address recognition, and acknowledgment.

### STATUS DECODER AND STATUS REGISTER

The status decoder takes all of the internal status bits and compresses them into a 5-bit code. This code is unique for each I<sup>2</sup>C bus status. The 5-bit code may be used to generate vector addresses for fast processing of the various service routines. Each service routine processes a particular bus status. There are 26 possible bus states if all four modes of SIO1 are used. The 5-bit status code is latched into the five most significant bits of the status register when the serial interrupt flag is set (by hardware) and remains stable until the interrupt flag is cleared by software. The three least significant bits of the status register are always zero. If the status code is used as a vector to service routines, then the routines are displaced by eight address locations. Eight bytes of code is sufficient for most of the service routines (see the software example in this section).

### The Four SIO1 Special Function Registers

The microcontroller interfaces to SIO1 via four special function registers. These four SFRs (S1ADR, S1DAT, S1CON, and S1STA) are described individually in the following sections.

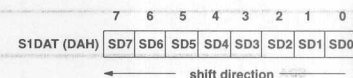
**The Address Register, S1ADR:** The CPU can read from and write to this 8-bit, directly addressable SFR. S1ADR is not affected by the SIO1 hardware. The contents of this register are irrelevant when SIO1 is in a master mode. In the slave modes, the seven most significant bits must be loaded with the microcontroller's own slave address, and, if the least significant bit is set, the general call address (00H) is recognized; otherwise it is ignored.

	7	6	5	4	3	2	1	0
S1ADR (DBH)	X	X	X	X	X	X	X	GC
	own slave address							

The most significant bit corresponds to the first bit received from the I<sup>2</sup>C bus after a start

condition. A logic 1 in S1ADR corresponds to a high level on the I<sup>2</sup>C bus, and a logic 0 corresponds to a low level on the bus.

**The Data Register, S1DAT:** S1DAT contains a byte of serial data to be transmitted or a byte which has just been received. The CPU can read from and write to this 8-bit, directly addressable SFR while it is not in the process of shifting a byte. This occurs when SIO1 is in a defined state and the serial interrupt flag is set. Data in S1DAT remains stable as long as SI is set. Data in S1DAT is always shifted from right to left: the first bit to be transmitted is the MSB (bit 7), and, after a byte has been received, the first bit of received data is located at the MSB of S1DAT. While data is being shifted out, data on the bus is simultaneously being shifted in; S1DAT always contains the last data byte present on the bus. Thus, in the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data in S1DAT.



### SD7 - SD0:

Eight bits to be transmitted or just received. A logic 1 in S1DAT corresponds to a high level on the I<sup>2</sup>C bus, and a logic 0 corresponds to a low level on the bus. Serial data shifts through S1DAT from right to left. Figure 15 shows how data in S1DAT is serially transferred to and from the SDA line.

S1DAT and the ACK flag form a 9-bit shift register which shifts in or shifts out an 8-bit byte, followed by an acknowledge bit. The ACK flag is controlled by the SIO1 hardware and cannot be accessed by the CPU. Serial data is shifted through the ACK flag into S1DAT on the rising edges of serial clock pulses on the SCL line. When a byte has been shifted into S1DAT, the serial data is available in S1DAT, and the acknowledge bit is returned by the control logic during the ninth clock pulse. Serial data is shifted out from S1DAT via a buffer (BSD7) on the falling edges of clock pulses on the SCL line.

When the CPU writes to S1DAT, BSD7 is loaded with the content of S1DAT.7, which is the first bit to be transmitted to the SDA line (see Figure 16). After nine serial clock pulses, the eight bits in S1DAT will have been transmitted to the SDA line, and the acknowledge bit will be present in ACK. Note that the eight transmitted bits are shifted back into S1DAT.

## 80C51 Family Derivatives

## 8XC552/562 overview

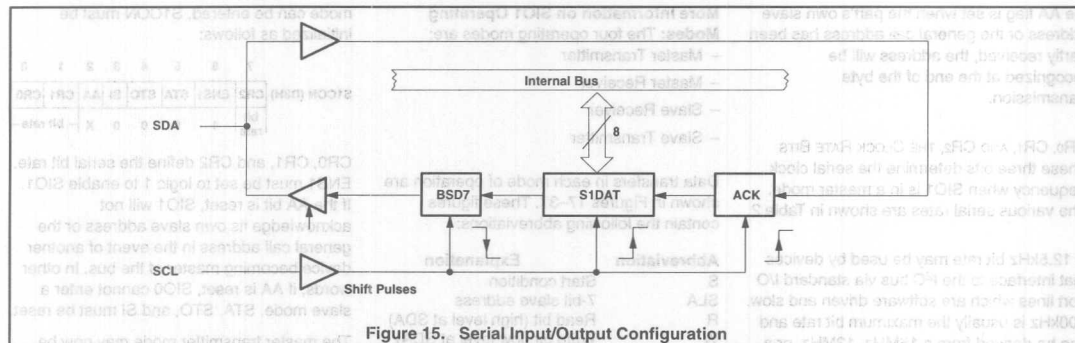


Figure 15. Serial Input/Output Configuration

**The Control Register, S1CON:** The CPU can read from and write to this 8-bit, directly addressable SFR. Two bits are affected by the SIO1 hardware: the SI bit is set when a serial interrupt is requested, and the STO bit is cleared when a STOP condition is present on the I<sup>2</sup>C bus. The STO bit is also cleared when ENS1 = "0".

	7	6	5	4	3	2	1	0
S1CON (D8H)	CR2	ENS1	STA	STO	SI	AA	CR1	CR0

**ENS1, THE SIO1 ENABLE BIT**

ENS1 = "0": When ENS1 is "0", the SDA and SCL outputs are in a high impedance state. SDA and SCL input signals are ignored, SIO1 is in the "not addressed" slave state, and the STO bit in S1CON is forced to "0". No other bits are affected. P1.6 and P1.7 may be used as open drain I/O ports.

ENS1 = "1": When ENS1 is "1", SIO1 is enabled. The P1.6 and P1.7 port latches must be set to logic 1.

ENS1 should not be used to temporarily release SIO1 from the I<sup>2</sup>C bus since, when ENS1 is reset, the I<sup>2</sup>C bus status is lost. The AA flag should be used instead (see description of the AA flag in the following text).

In the following text, it is assumed that ENS1 = "1".

**STA, THE START FLAG**

STA = "1": When the STA bit is set to enter a master mode, the SIO1 hardware checks the status of the I<sup>2</sup>C bus and generates a START condition if the bus is free. If the bus is not free, then SIO1 waits for a STOP condition (which will free the bus) and generates a START condition after a delay of a half clock period of the internal serial clock generator.

If STA is set while SIO1 is already in a master mode and one or more bytes are transmitted or received, SIO1 transmits a repeated START condition. STA may be set at any

time. STA may also be set when SIO1 is an addressed slave.

STA = "0": When the STA bit is reset, no START condition or repeated START condition will be generated.

**STO, THE STOP FLAG**

STO = "1": When the STO bit is set while SIO1 is in a master mode, a STOP condition is transmitted to the I<sup>2</sup>C bus. When the STOP condition is detected on the bus, the SIO1 hardware clears the STO flag. In a slave mode, the STO flag may be set to recover from an error condition. In this case, no STOP condition is transmitted to the I<sup>2</sup>C bus. However, the SIO1 hardware behaves as if a STOP condition has been received and switches to the defined "not addressed" slave receiver mode. The STO flag is automatically cleared by hardware.

If the STA and STO bits are both set, the a STOP condition is transmitted to the I<sup>2</sup>C bus if SIO1 is in a master mode (in a slave mode, SIO1 generates an internal STOP condition which is not transmitted). SIO1 then transmits a START condition.

STO = "0": When the STO bit is reset, no STOP condition will be generated.

**SI, THE SERIAL INTERRUPT FLAG**

SI = "1": When the SI flag is set, then, if the EA and ES1 (interrupt enable register) bits are also set, a serial interrupt is requested. SI is set by hardware when one of 25 of the 26 possible SIO1 states is entered. The only state that does not cause SI to be set is state F8H, which indicates that no relevant state information is available.

While SI is set, the low period of the serial clock on the SCL line is stretched, and the serial transfer is suspended. A high level on the SCL line is unaffected by the serial interrupt flag. SI must be reset by software.

SI = "0": When the SI flag is reset, no serial interrupt is requested, and there is no stretching of the serial clock on the SCL line.

**AA, THE ASSERT ACKNOWLEDGE FLAG**

AA = "1": If the AA flag is set, an acknowledge (low level to SDA) will be returned during the acknowledge clock pulse on the SCL line when:

- The "own slave address" has been received
- The general call address has been received while the general call bit (GC) in S1ADR is set
- A data byte has been received while SIO1 is in the master receiver mode
- A data byte has been received while SIO1 is in the addressed slave receiver mode

AA = "0": If the AA flag is reset, a not acknowledge (high level to SDA) will be returned during the acknowledge clock pulse on SCL when:

- A data has been received while SIO1 is in the master receiver mode
- A data byte has been received while SIO1 is in the addressed slave receiver mode

When SIO1 is in the addressed slave transmitter mode, state C8H will be entered after the last serial is transmitted (see Figure 20). When SI is cleared, SIO1 leaves state C8H, enters the not addressed slave receiver mode, and the SDA line remains at a high level. In state C8H, the AA flag can be set again for future address recognition.

When SIO1 is in the not addressed slave mode, its own slave address and the general call address are ignored. Consequently, no acknowledge is returned, and a serial interrupt is not requested. Thus, SIO1 can be temporarily released from the I<sup>2</sup>C bus while the bus status is monitored. While SIO1 is released from the bus, START and STOP conditions are detected, and serial data is shifted in. Address recognition can be resumed at any time by setting the AA flag. If



## 80C51 Family Derivatives

## 8XC552/562 overview

the AA flag is set when the part's own slave address or the general call address has been partly received, the address will be recognized at the end of the byte transmission.

**CR0, CR1, AND CR2, THE CLOCK RATE BITS**  
These three bits determine the serial clock frequency when SIO1 is in a master mode. The various serial rates are shown in Table 2.

A 12.5kHz bit rate may be used by devices that interface to the I<sup>2</sup>C bus via standard I/O port lines which are software driven and slow. 100kHz is usually the maximum bit rate and can be derived from a 16MHz, 12MHz, or a 6MHz oscillator. A variable bit rate (0.5kHz to 62.5kHz) may also be used if Timer 1 is not required for any other purpose while SIO1 is in a master mode.

The frequencies shown in Table 2 are unimportant when SIO1 is in a slave mode. In the slave modes, SIO1 will automatically synchronize with any clock frequency up to 100kHz.

**The Status Register, S1STA:** S1STA is an 8-bit read-only special function register. The three least significant bits are always zero. The five most significant bits contain the status code. There are 26 possible status codes. When S1STA contains F8H, no relevant state information is available and no serial interrupt is requested. All other S1STA values correspond to defined SIO1 states. When each of these states is entered, a serial interrupt is requested (SI = "1"). A valid status code is present in S1STA one machine cycle after SI is set by hardware and is still present one machine cycle after SI has been reset by software.

### More Information on SIO1 Operating

**Modes:** The four operating modes are:

- Master Transmitter
- Master Receiver
- Slave Receiver
- Slave Transmitter

Data transfers in each mode of operation are shown in Figures 17–37. These figures contain the following abbreviations:

Abbreviation	Explanation
S	Start condition
SLA	7-bit slave address
R	Read bit (high level at SDA)
W	Write bit (low level at SDA)
A	Acknowledge bit (low level at SDA)
$\bar{A}$	Not acknowledge bit (high level at SDA)
Data	8-bit data byte
P	Stop condition

In Figures 17–37, circles are used to indicate when the serial interrupt flag is set. The numbers in the circles show the status code held in the S1STA register. At these points, a service routine must be executed to continue or complete the serial transfer. These service routines are not critical since the serial transfer is suspended until the serial interrupt flag is cleared by software.

When a serial interrupt routine is entered, the status code in S1STA is used to branch to the appropriate service routine. For each status code, the required software action and details of the following serial transfer are given in Tables 3–7.

**Master Transmitter Mode:** In the master transmitter mode, a number of data bytes are transmitted to a slave receiver (see Figure 17). Before the master transmitter

mode can be entered, S1CON must be initialized as follows:

	7	6	5	4	3	2	1	0
S1CON (D8H)	CR2	ENS1	STA	STO	SI	AA	CR1	CR0
bit rate	1	0	0	0	0	X		

CR0, CR1, and CR2 define the serial bit rate. ENS1 must be set to logic 1 to enable SIO1. If the AA bit is reset, SIO1 will not acknowledge its own slave address or the general call address in the event of another device becoming master of the bus. In other words, if AA is reset, SIO0 cannot enter a slave mode. STA, STO, and SI must be reset.

The master transmitter mode may now be entered by setting the STA bit using the SETB instruction. The SIO1 logic will now test the I<sup>2</sup>C bus and generate a start condition as soon as the bus becomes free. When a START condition is transmitted, the serial interrupt flag (SI) is set, and the status code in the status register (S1STA) will be 08H. This status code must be used to vector to an interrupt service routine that loads S1DAT with the slave address and the data direction bit (SLA+W). The SI bit in S1CON must then be reset before the serial transfer can continue.

When the slave address and the direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag (SI) is set again, and a number of status codes in S1STA are possible. There are 18H, 20H, or 38H for the master mode and also 68H, 78H, or 80H if the slave mode was enabled (AA = logic 1). The appropriate action to be taken for each of these status codes is detailed in Table 3. After a repeated start condition (state 10H), SIO1 may switch to the master receiver mode by loading S1DAT with SLA+R).

# 80C51 Family Derivatives

## 8XC552/562 overview

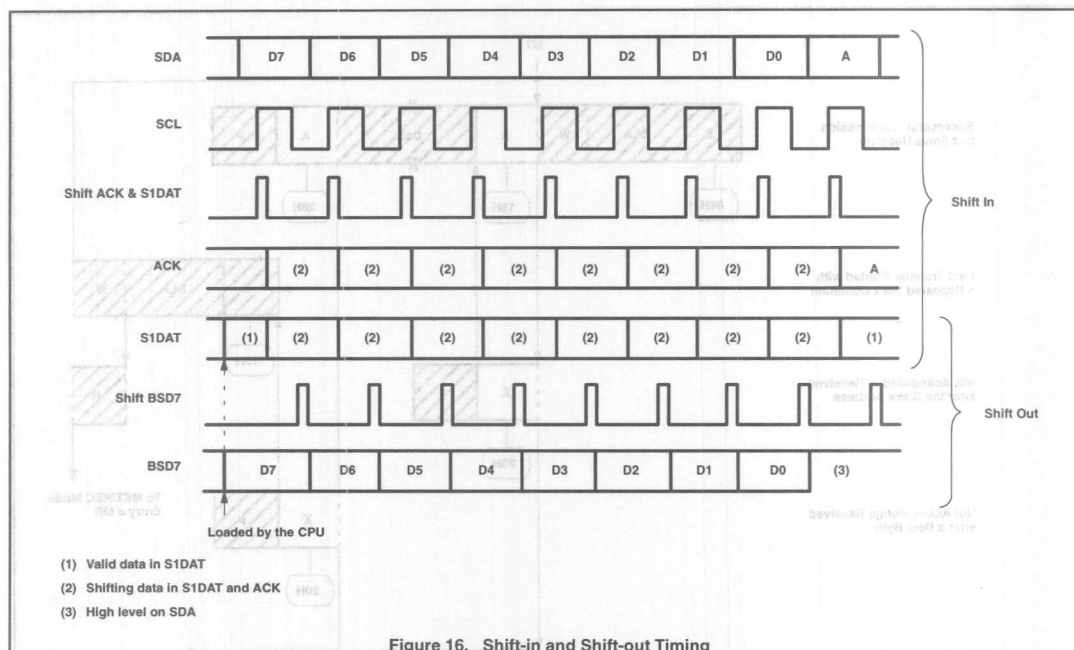
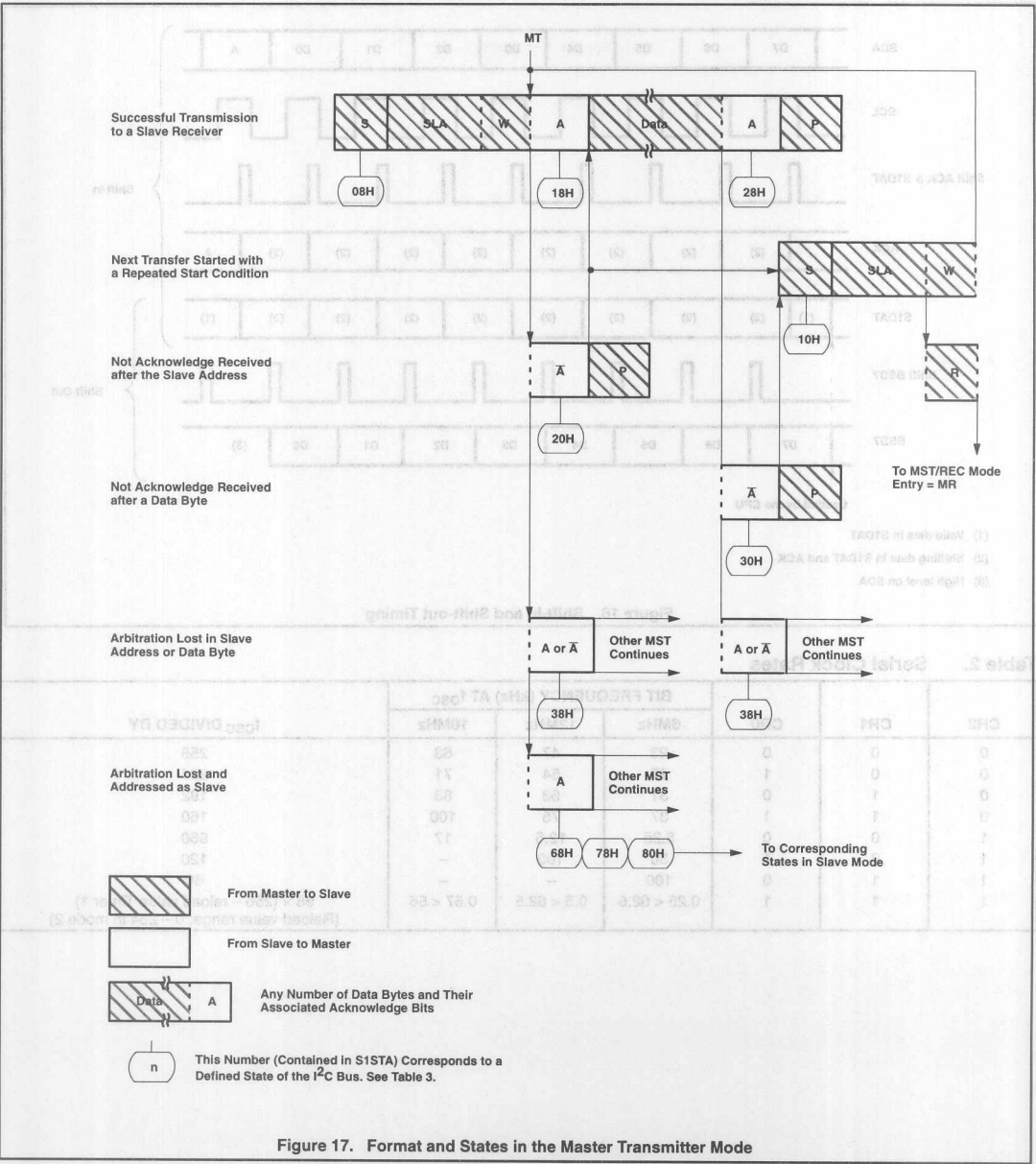


Table 2. Serial Clock Rates

CR2	CR1	CR0	BIT FREQUENCY (kHz) AT $f_{osc}$			$f_{osc}$ DIVIDED BY
			6MHz	12MHz	16MHz	
0	0	0	23	47	63	256
0	0	1	27	54	71	224
0	1	0	31	63	83	192
0	1	1	37	75	100	160
1	0	0	6.25	12.5	17	960
1	0	1	50	100	—	120
1	1	0	100	—	—	60
1	1	1	0.25 < 62.5	0.5 < 62.5	0.67 < 56	96 × (256 – reload value Timer 1) (Reload value range: 0 – 254 in mode 2)

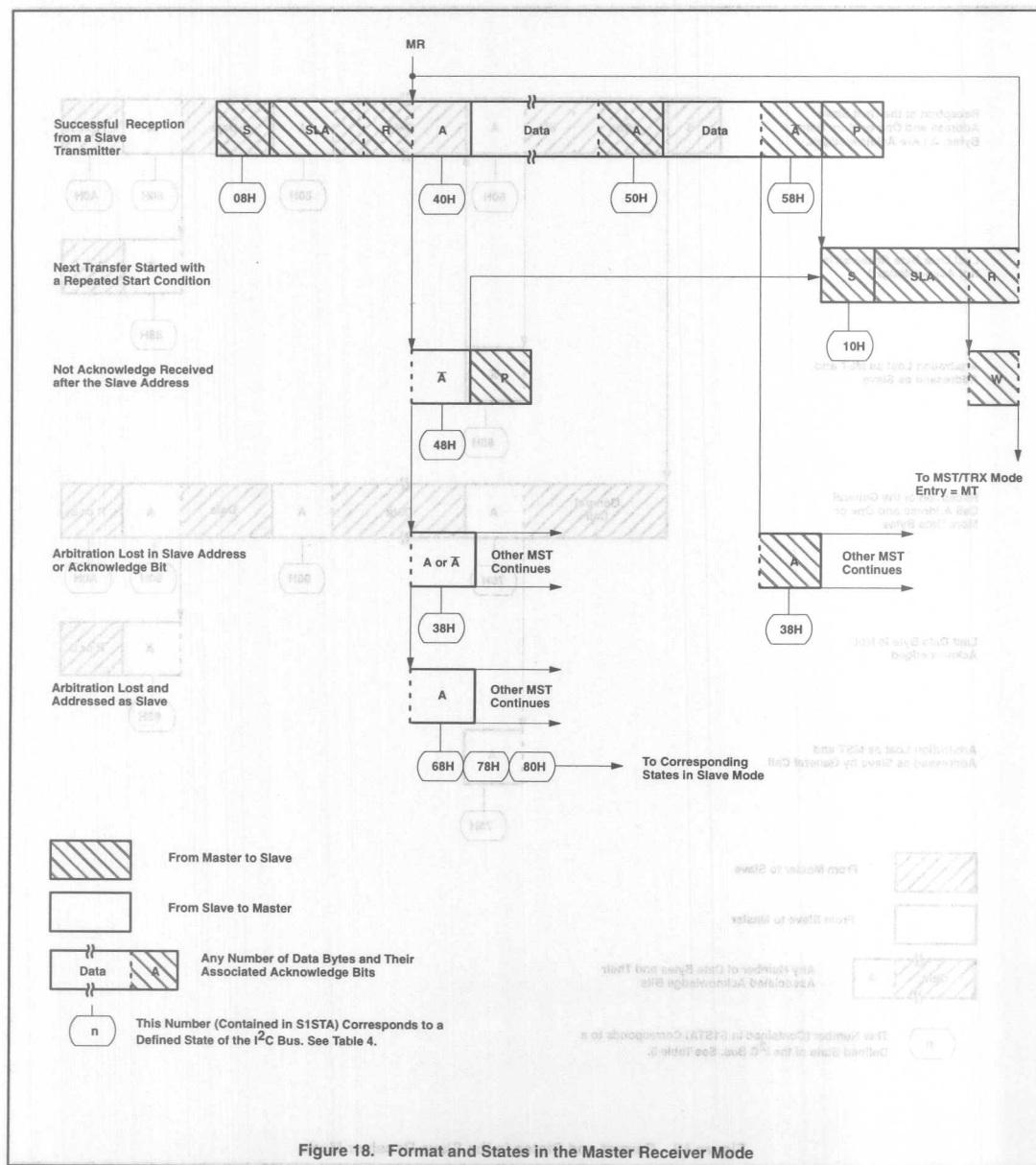
80C51 Family Derivatives

8XC552/562 overview



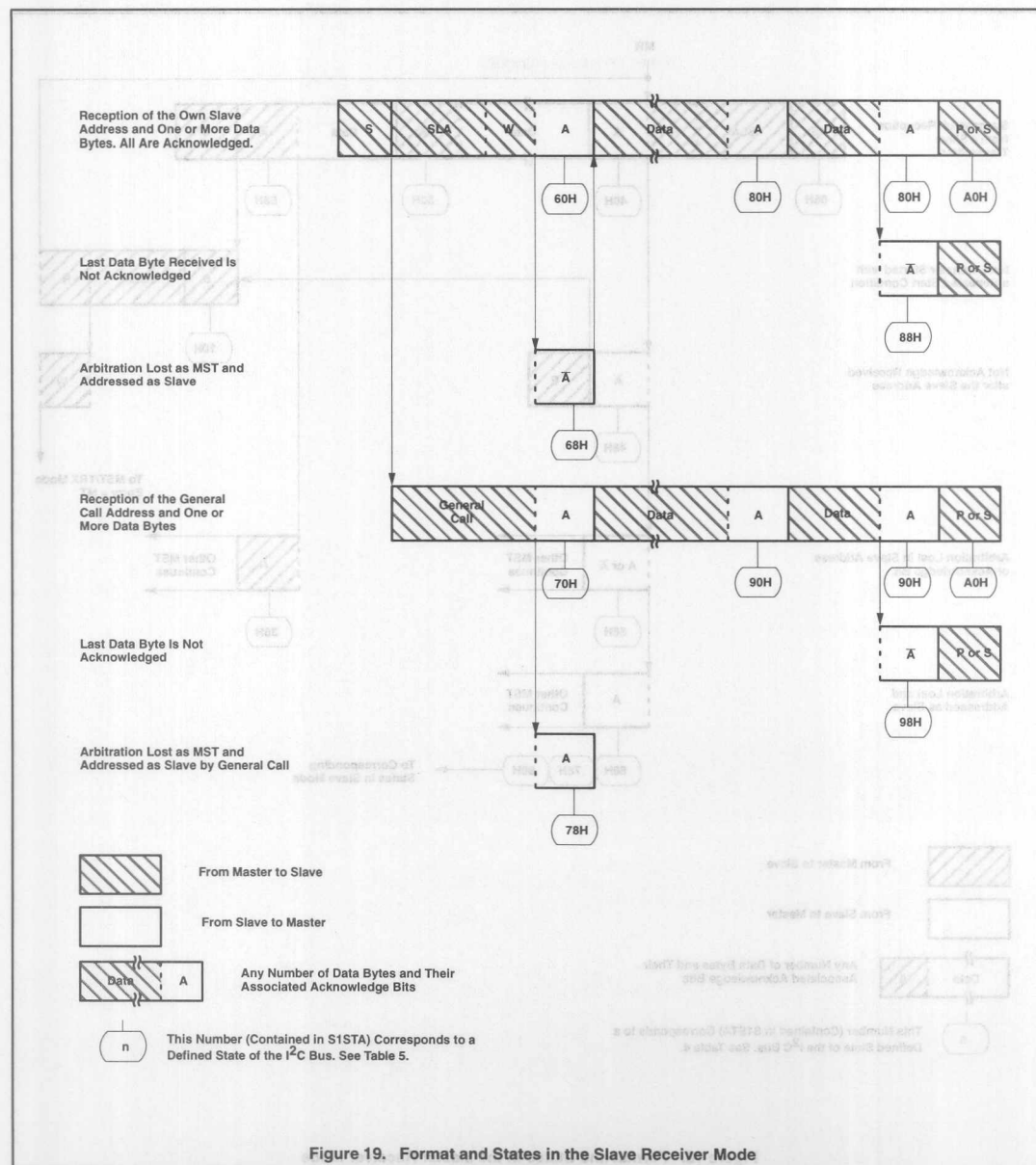
## 80C51 Family Derivatives

## 8XC552/562 overview



## 80C51 Family Derivatives

## 8XC552/562 overview





## 80C51 Family Derivatives

## 8XC552/562 overview

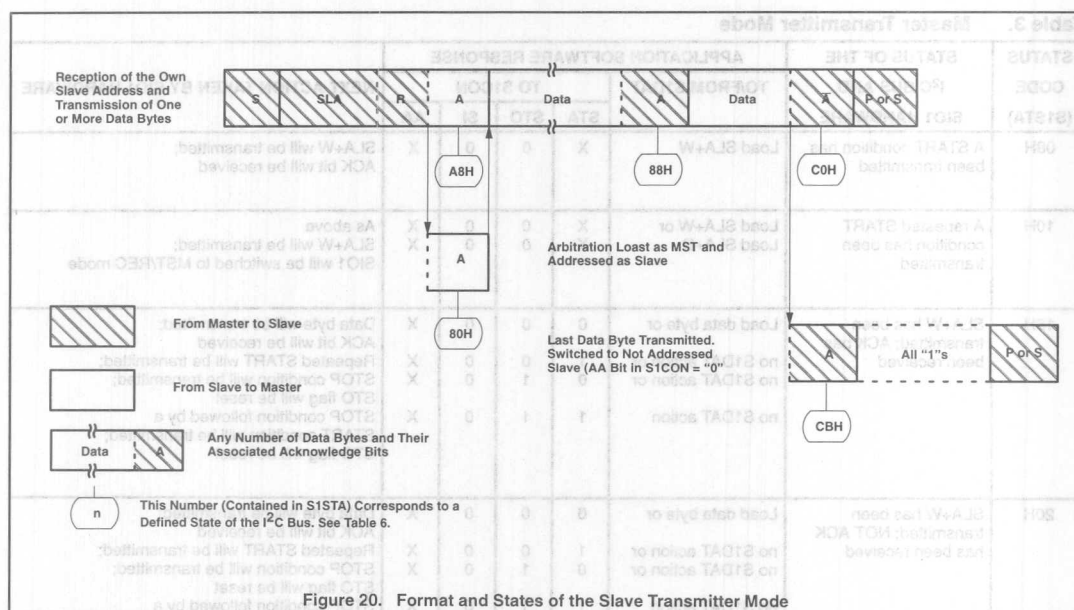


Figure 20. Format and States of the Slave Transmitter Mode

**Master Receiver Mode:** In the master receiver mode, a number of data bytes are received from a slave transmitter (see Figure 18). The transfer is initialized as in the master transmitter mode. When the start condition has been transmitted, the interrupt service routine must load S1DAT with the 7-bit slave address and the data direction bit (SLA+R). The SI bit in S1CON must then be cleared before the serial transfer can continue.

When the slave address and the data direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag (SI) is set again, and a number of status codes in S1STA are possible. These are 40H, 48H, or 38H for the master mode and also 68H, 78H, or B0H if the slave mode was enabled (AA = logic 1). The appropriate action to be taken for each of these status codes is detailed in Table 4. ENS1, CR1, and CR0 are not affected by the serial transfer and are not referred to in Table 4. After a repeated start condition (state 10H), SIO1 may switch to the master transmitter mode by loading S1DAT with SLA+W.

**Slave Receiver Mode:** In the slave receiver mode, a number of data bytes are received from a master transmitter (see Figure 19). To initiate the slave receiver mode, S1ADR and S1CON must be loaded as follows:

	7	6	5	4	3	2	1	0
S1ADR (DBH)	X	X	X	X	X	X	X	GC
	own slave address							

The upper 7 bits are the address to which SIO1 will respond when addressed by a master. If the LSB (GC) is set, SIO1 will respond to the general call address (00H); otherwise it ignores the general call address.

	7	6	5	4	3	2	1	0
S1CON (DBH)	CR2	ENS1	STA	STO	SI	AA	CR1	CR0
	X	1	0	0	0	1	X	X

CR0, CR1, and CR12 do not affect SIO1 in the slave mode. ENS1 must be set to logic 1 to enable SIO1. The AA bit must be set to enable SIO1 to acknowledge its own slave address or the general call address. STA, STO, and SI must be reset.

When S1ADR and S1CON have been initialized, SIO1 waits until it is addressed by its own slave address followed by the data direction bit which must be "0" (W) for SIO1 to operate in the slave receiver mode. After its own slave address and the W bit have been received, the serial interrupt flag (I) is set and a valid status code can be read from S1STA. This status code is used to vector to an interrupt service routine, and the appropriate action to be taken for each of these status codes is detailed in Table 5. The slave receiver mode may also be entered if arbitration is lost while SIO1 is in the master mode (see status 68H and 78H).

If the AA bit is reset during a transfer, SIO1 will return a not acknowledge (logic 1) to SDA after the next received data byte. While AA is reset, SIO1 does not respond to its own slave address or a general call address. However, the I<sup>2</sup>C bus is still monitored and address recognition may be resumed at any time by setting AA. This means that the AA bit may be used to temporarily isolate SIO1 from the I<sup>2</sup>C bus.

## 80C51 Family Derivatives

## 8XC552/562 overview

Table 3. Master Transmitter Mode

STATUS CODE (S1STA)	STATUS OF THE I <sup>2</sup> C BUS AND SIO1 HARDWARE	APPLICATION SOFTWARE RESPONSE				NEXT ACTION TAKEN BY SIO1 HARDWARE	
		TO/FROM S1DAT	TO S1CON				
			STA	STO	SI	AA	
08H	A START condition has been transmitted	Load SLA+W	X	0	0	X	SLA+W will be transmitted; ACK bit will be received
10H	A repeated START condition has been transmitted	Load SLA+W or Load SLA+R	X X	0 0	0 0	X X	As above SLA+W will be transmitted; SIO1 will be switched to MST/REC mode
18H	SLA+W has been transmitted; ACK has been received	Load data byte or	0	0	0	X	Data byte will be transmitted; ACK bit will be received
		no S1DAT action or	1	0	0	X	Repeated START will be transmitted; STOP condition will be transmitted; STO flag will be reset
		no S1DAT action or	0	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
20H	SLA+W has been transmitted; NOT ACK has been received	Load data byte or	0	0	0	X	Data byte will be transmitted; ACK bit will be received
		no S1DAT action or	1	0	0	X	Repeated START will be transmitted; STOP condition will be transmitted; STO flag will be reset
		no S1DAT action or	0	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
28H	Data byte in S1DAT has been transmitted; ACK has been received	Load data byte or	0	0	0	X	Data byte will be transmitted; ACK bit will be received
		no S1DAT action or	1	0	0	X	Repeated START will be transmitted; STOP condition will be transmitted; STO flag will be reset
		no S1DAT action or	0	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
30H	Data byte in S1DAT has been transmitted; NOT ACK has been received	Load data byte or	0	0	0	X	Data byte will be transmitted; ACK bit will be received
		no S1DAT action or	1	0	0	X	Repeated START will be transmitted; STOP condition will be transmitted; STO flag will be reset
		no S1DAT action or	0	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
38H	Arbitration lost in SLA+R/W or Data bytes	No S1DAT action or	0	0	0	X	I <sup>2</sup> C bus will be released; not addressed slave will be entered
		No S1DAT action	1	0	0	X	A START condition will be transmitted when the bus becomes free

## 80C51 Family Derivatives

## 8XC552/562 overview

Table 4. Master Receiver Mode

STATUS CODE (S1STA)	STATUS OF THE I <sup>2</sup> C BUS AND SIO1 HARDWARE	APPLICATION SOFTWARE RESPONSE					NEXT ACTION TAKEN BY SIO1 HARDWARE
		TO/FROM S1DAT	TO S1CON	TO S1CON	SI	AA	
08H	A START condition has been transmitted	Load SLA+R	X	0	0	X	SLA+R will be transmitted; ACK bit will be received
10H	A repeated START condition has been transmitted	Load SLA+R or Load SLA+W	X X	0 0	0 0	X X	As above SLA+W will be transmitted; SIO1 will be switched to MST/TRX mode
38H	Arbitration lost in NOT ACK bit	No S1DAT action or No S1DAT action	0 1	0 0	0 0	X X	I <sup>2</sup> C bus will be released; SIO1 will enter a slave mode A START condition will be transmitted when the bus becomes free
40H	SLA+R has been transmitted; ACK has been received	No S1DAT action or no S1DAT action	0 0	0 0	0 0	0 1	Data byte will be received; NOT ACK bit will be returned Data byte will be received; ACK bit will be returned
48H	SLA+R has been transmitted; NOT ACK has been received	No S1DAT action or no S1DAT action or no S1DAT action	1 0 1	0 1 1	0 0 0	X X X	Repeated START condition will be transmitted STOP condition will be transmitted; STO flag will be reset STOP condition followed by a START condition will be transmitted; STO flag will be reset
50H	Data byte has been received; ACK has been returned	Read data byte or read data byte	0 0	0 0	0 0	0 1	Data byte will be received; NOT ACK bit will be returned Data byte will be received; ACK bit will be returned
58H	Data byte has been received; NOT ACK has been returned	Read data byte or read data byte or read data byte	1 0 1	0 1 1	0 0 0	X X X	Repeated START condition will be transmitted STOP condition will be transmitted; STO flag will be reset STOP condition followed by a START condition will be transmitted; STO flag will be reset

## 80C51 Family Derivatives

## 8XC552/562 overview

Table 5. Slave Receiver Mode

STATUS CODE (S1STA)	STATUS OF THE I <sup>2</sup> C BUS AND SIO1 HARDWARE	APPLICATION SOFTWARE RESPONSE						NEXT ACTION TAKEN BY SIO1 HARDWARE
		TO/FROM S1DAT	TO S1CON					
			AA	STA	STO	SI	AA	
60H	Own SLA+W has been received; ACK has been returned	No S1DAT action or no S1DAT action	X X	0 0	0 0	0 1	Data byte will be received and NOT ACK will be returned Data byte will be received and ACK will be returned	
68H	Arbitration lost in SLA+R/W as master; Own SLA+W has been received, ACK returned	No S1DAT action or no S1DAT action	X X	0 0	0 0	0 1	Data byte will be received and NOT ACK will be returned Data byte will be received and ACK will be returned	
70H	General call address (00H) has been received; ACK has been returned	No S1DAT action or no S1DAT action	X X	0 0	0 0	0 1	Data byte will be received and NOT ACK will be returned Data byte will be received and ACK will be returned	
78H	Arbitration lost in SLA+R/W as master; General call address has been received, ACK has been returned	No S1DAT action or no S1DAT action	X X	0 0	0 0	0 1	Data byte will be received and NOT ACK will be returned Data byte will be received and ACK will be returned	
80H	Previously addressed with own SLV address; DATA has been received; ACK has been returned	Read data byte or read data byte	X X	0 0	0 0	0 1	Data byte will be received and NOT ACK will be returned Data byte will be received and ACK will be returned	
88H	Previously addressed with own SLA; DATA byte has been received; NOT ACK has been returned	Read data byte or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1 Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.	
		read data byte or	0	0	0	1		
		read data byte or	1	0	0	0		
		read data byte	1	0	0	1		
90H	Previously addressed with General Call; DATA byte has been received; ACK has been returned	Read data byte or read data byte	X X	0 0	0 0	0 1	Data byte will be received and NOT ACK will be returned Data byte will be received and ACK will be returned	
98H	Previously addressed with General Call; DATA byte has been received; NOT ACK has been returned	Read data byte or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1 Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.	
		read data byte or	0	0	0	1		
		read data byte or	1	0	0	0		
		read data byte	1	0	0	1		

## 80C51 Family Derivatives

## 8XC552/562 overview

Table 5. Slave Receiver Mode (Continued)

STATUS CODE (S1STA)	STATUS OF THE I <sup>2</sup> C BUS AND SIO1 HARDWARE	APPLICATION SOFTWARE RESPONSE TO/FROM S1DAT	APPLICATION SOFTWARE RESPONSE TO S1CON				NEXT ACTION TAKEN BY SIO1 HARDWARE
			STA	STO	SI	AA	
A0H	A STOP condition or repeated START condition has been received while still addressed as SLV/REC or SLV/TRX	Read data byte or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free. Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.
		read data byte or	0	0	0	1	
		read data byte or	1	0	0	0	
		read data byte	1	0	0	1	

Table 6. Slave Transmitter Mode

STATUS CODE (S1STA)	STATUS OF THE I <sup>2</sup> C BUS AND SIO1 HARDWARE	APPLICATION SOFTWARE RESPONSE TO/FROM S1DAT	APPLICATION SOFTWARE RESPONSE TO S1CON				NEXT ACTION TAKEN BY SIO1 HARDWARE
			STA	STO	SI	AA	
A8H	Own SLA+R has been received; ACK has been returned	Load data byte or	X	0	0	0	Last data byte will be transmitted and ACK bit will be received. Data byte will be transmitted; ACK will be received.
		load data byte	X	0	0	1	
B0H	Arbitration lost in SLA+R/W as master; Own SLA+R has been received, ACK has been returned	Load data byte or	X	0	0	0	Last data byte will be transmitted and ACK bit will be received. Data byte will be transmitted; ACK bit will be received.
		load data byte	X	0	0	1	
B8H	Data byte in S1DAT has been transmitted; ACK has been received	Load data byte or	X	0	0	0	Last data byte will be transmitted and ACK bit will be received. Data byte will be transmitted; ACK bit will be received.
		load data byte	X	0	0	1	
C0H	Data byte in S1DAT has been transmitted; NOT ACK has been received	No S1DAT action or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free. Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.
		no S1DAT action or	0	0	0	1	
		no S1DAT action or	1	0	0	0	
		no S1DAT action	1	0	0	1	
C8H	Last data byte in S1DAT has been transmitted (AA = 0); ACK has been received	No S1DAT action or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free. Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.
		no S1DAT action or	0	0	0	1	
		no S1DAT action or	1	0	0	0	
		no S1DAT action	1	0	0	1	



## 80C51 Family Derivatives

## 8XC552/562 overview

**Slave Transmitter Mode:** In the slave transmitter mode, a number of data bytes are transmitted to a master receiver (see Figure 20). Data transfer is initialized as in the slave receiver mode. When S1ADR and S1CON have been initialized, SIO1 waits until it is addressed by its own slave address followed by the data direction bit which must be "1" (R) for SIO1 to operate in the slave transmitter mode. After its own slave address and the R bit have been received, the serial interrupt flag (SI) is set and a valid status code can be read from S1STA. This status code is used to vector to an interrupt service routine, and the appropriate action to be taken for each of these status codes is detailed in Table 6. The slave transmitter mode may also be entered if arbitration is lost while SIO1 is in the master mode (see state B0H).

If the AA bit is reset during a transfer, SIO1 will transmit the last byte of the transfer and enter state C0H or C8H. SIO1 is switched to the not addressed slave mode and will ignore the master receiver if it continues the transfer. Thus the master receiver receives all 1s as serial data. While AA is reset, SIO1 does not respond to its own slave address or a general call address. However, the I<sup>2</sup>C bus is still monitored, and address recognition may be resumed at any time by setting AA. This means that the AA bit may be used to temporarily isolate SIO1 from the I<sup>2</sup>C bus.

**Miscellaneous States:** There are two S1STA codes that do not correspond to a defined SIO1 hardware state (see Table 7). These are discussed below.

**S1STA = F8H:**

This status code indicates that no relevant information is available because the serial interrupt flag, SI, is not yet set. This occurs between other states and when SIO1 is not involved in a serial transfer.

**S1STA = 00H:**

This status code indicates that a bus error has occurred during an SIO1 serial transfer. A bus error is caused when a START or STOP condition occurs at an illegal position in the format frame. Examples of such illegal positions are during the serial transfer of an address byte, a data byte, or an acknowledge bit. A bus error may also be caused when external interference disturbs the internal SIO1 signals. When a bus error occurs, SI is set. To recover from a bus error, the STO flag must be set and SI must be cleared. This causes SIO1 to enter the "not addressed" slave mode (a defined state) and to clear the

STO flag (no other bits in S1CON are affected). The SDA and SCL lines are released (a STOP condition is not transmitted).

**Some Special Cases:** The SIO1 hardware has facilities to handle the following special cases that may occur during a serial transfer:

**Simultaneous Repeated START Conditions from Two Masters**

A repeated START condition may be generated in the master transmitter or master receiver modes. A special case occurs if another master simultaneously generates a repeated START condition (see Figure 21). Until this occurs, arbitration is not lost by either master since they were both transmitting the same data.

If the SIO1 hardware detects a repeated START condition on the I<sup>2</sup>C bus before generating a repeated START condition itself, it will release the bus, and no interrupt request is generated. If another master frees the bus by generating a STOP condition, SIO1 will transmit a normal START condition (state 08H), and a retry of the total serial data transfer can commence.

**DATA TRANSFER AFTER LOSS OF ARBITRATION**  
Arbitration may be lost in the master transmitter and master receiver modes (see Figure 13). Loss of arbitration is indicated by the following states in S1STA: 38H, 68H, 78H, and B0H (see Figures 17 and 18).

If the STA flag in S1CON is set by the routines which service these states, then, if the bus is free again, a START condition (state 08H) is transmitted without intervention by the CPU, and a retry of the total serial transfer can commence.

**FORCED ACCESS TO THE I<sup>2</sup>C BUS**

In some applications, it may be possible for an uncontrolled source to cause a bus hang-up. In such situations, the problem may be caused by interference, temporary interruption of the bus or a temporary short-circuit between SDA and SCL.

If an uncontrolled source generates a superfluous START or masks a STOP condition, then the I<sup>2</sup>C bus stays busy indefinitely. If the STA flag is set and bus access is not obtained within a reasonable amount of time, then a forced access to the I<sup>2</sup>C bus is possible. This is achieved by setting the STO flag while the STA flag is still set. No STOP condition is transmitted. The SIO1 hardware behaves as if a STOP

condition was received and is able to transmit a START condition. The STO flag is cleared by hardware (see Figure 22).

**I<sup>2</sup>C BUS OBSTRUCTED BY A LOW LEVEL ON SCL OR SDA**

An I<sup>2</sup>C bus hang-up occurs if SDA or SCL is pulled LOW by an uncontrolled source. If the SCL line is obstructed (pulled LOW) by a device on the bus, no further serial transfer is possible, and the SIO1 hardware cannot resolve this type of problem. When this occurs, the problem must be resolved by the device that is pulling the SCL bus line LOW.

If the SDA line is obstructed by another device on the bus (e.g., a slave device out of bit synchronization), the problem can be solved by transmitting additional clock pulses on the SCL line (see Figure 23). The SIO1 hardware transmits additional clock pulses when the STA flag is set, but no START condition can be generated because the SDA line is pulled LOW while the I<sup>2</sup>C bus is considered free. The SIO1 hardware attempts to generate a START condition after every two additional clock pulses on the SCL line. When the SDA line is eventually released, a normal START condition is transmitted, state 08H is entered, and the serial transfer continues.

If a forced bus access occurs or a repeated START condition is transmitted while SDA is obstructed (pulled LOW), the SIO1 hardware performs the same action as described above. In each case, state 08H is entered after a successful START condition is transmitted and normal serial transfer continues. Note that the CPU is not involved in solving these bus hang-up problems.

**BUS ERROR**

A bus error occurs when a START or STOP condition is present at an illegal position in the format frame. Examples of illegal positions are during the serial transfer of an address byte, a data or an acknowledge bit.

The SIO1 hardware only reacts to a bus error when it is involved in a serial transfer either as a master or an addressed slave. When a bus error is detected, SIO1 immediately switches to the not addressed slave mode, releases the SDA and SCL lines, sets the interrupt flag, and loads the status register with 00H. This status code may be used to vector to a service routine which either attempts the aborted serial transfer again or simply recovers from the error condition as shown in Table 7.

80C51 Family Derivatives

80C51 Family Derivatives

Table 7. Miscellaneous States

STATUS CODE (S1STA)	STATUS OF THE I <sup>2</sup> C BUS AND SIO1 HARDWARE	APPLICATION SOFTWARE RESPONSE					NEXT ACTION TAKEN BY SIO1 HARDWARE
		TO/FROM S1DAT	TO S1CON				
			STA	STO	SI	AA	
F8H	No relevant state information available; SI = 0	No S1DAT action	No S1CON action				Wait or proceed current transfer
00H	Bus error during MST or selected slave modes, due to an illegal START or STOP condition. State 00H can also occur when interference causes SIO1 to enter an undefined state.	No S1DAT action	0	1	0	X	Only the internal hardware is affected in the MST or addressed SLV modes. In all cases, the bus is released and SIO1 is switched to the not addressed SLV mode. STO is reset.

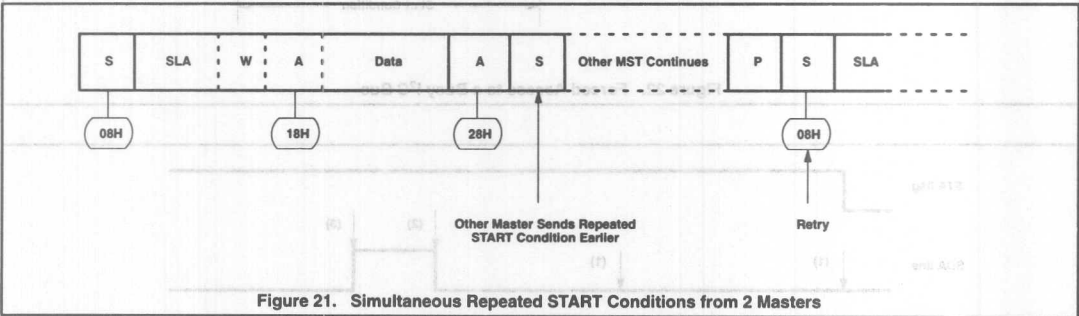
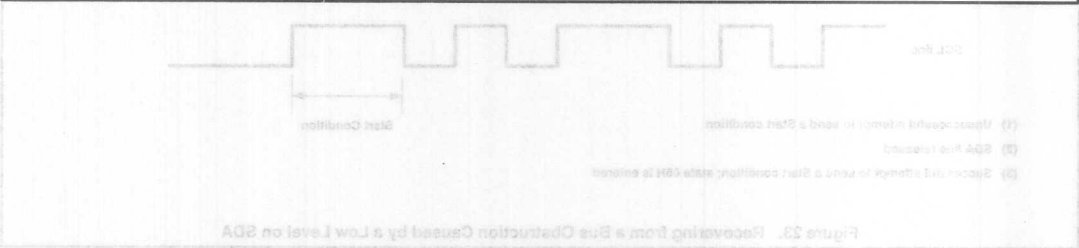


Figure 21. Simultaneous Repeated START Conditions from 2 Masters



Initialization routine: This routine is used to initialize the SIO1 hardware. It is called by the master routine. The routine consists of the following steps:

- 1. The SIO1 hardware is initialized by setting the SIO1 control register (SIO1CR) to 0x00.
- 2. The SIO1 hardware is initialized by setting the SIO1 status register (SIO1SR) to 0x00.
- 3. The SIO1 hardware is initialized by setting the SIO1 data register (SIO1DR) to 0x00.
- 4. The SIO1 hardware is initialized by setting the SIO1 interrupt register (SIO1IR) to 0x00.
- 5. The SIO1 hardware is initialized by setting the SIO1 error register (SIO1ER) to 0x00.
- 6. The SIO1 hardware is initialized by setting the SIO1 mode register (SIO1MR) to 0x00.
- 7. The SIO1 hardware is initialized by setting the SIO1 slave address register (SIO1SLA) to 0x00.
- 8. The SIO1 hardware is initialized by setting the SIO1 slave address register (SIO1SLA) to 0x00.
- 9. The SIO1 hardware is initialized by setting the SIO1 slave address register (SIO1SLA) to 0x00.
- 10. The SIO1 hardware is initialized by setting the SIO1 slave address register (SIO1SLA) to 0x00.

The SIO1 hardware is now ready for use. The master routine can now send data to the SIO1 hardware. The SIO1 hardware will receive the data and store it in the SIO1 data register (SIO1DR). The master routine can then read the data from the SIO1 data register (SIO1DR).

## 80C51 Family Derivatives

## 8XC552/562 overview

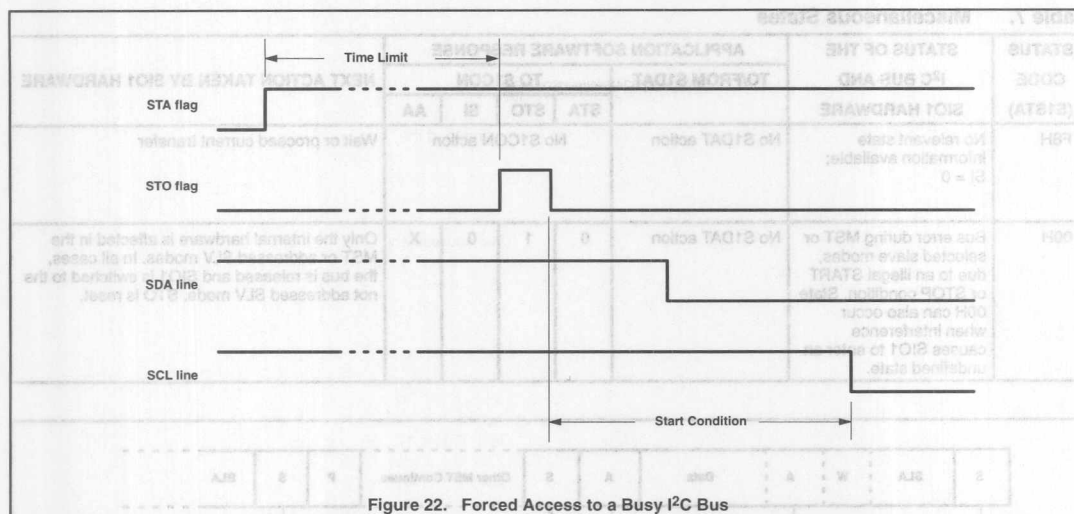
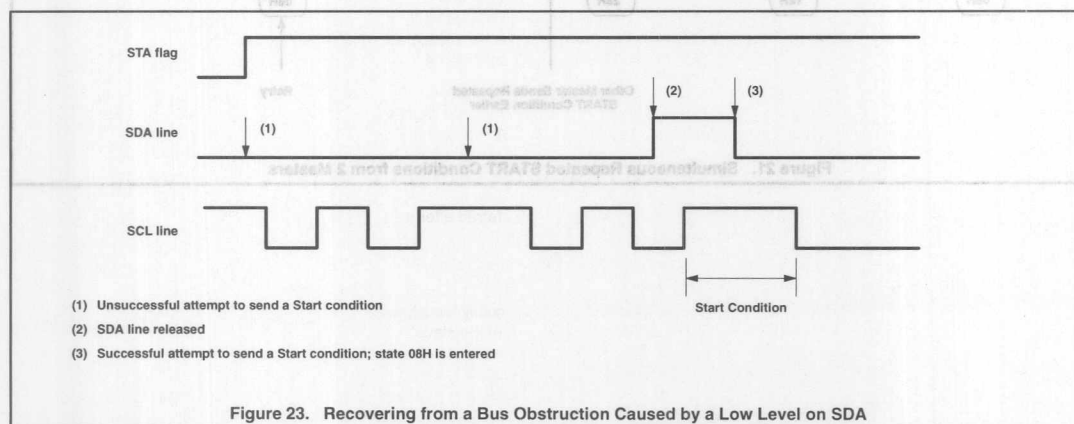
Figure 22. Forced Access to a Busy I<sup>2</sup>C Bus

Figure 23. Recovering from a Bus Obstruction Caused by a Low Level on SDA

## Software Examples of SIO1 Service

**Routines:** This section consists of a software example for:

- Initialization of SIO1 after a RESET
- Entering the SIO1 interrupt routine
- The 26 state service routines for the
  - Master transmitter mode
  - Master receiver mode
  - Slave receiver mode
  - Slave transmitter mode

## INITIALIZATION

In the initialization routine, SIO1 is enabled for both master and slave modes. For each mode, a number of bytes of internal data

RAM are allocated to the SIO to act as either a transmission or reception buffer. In this example, 8 bytes of internal data RAM are reserved for different purposes. The data memory map is shown in Figure 24. The initialization routine performs the following functions:

- S1ADR is loaded with the part's own slave address and the general call bit (GC)
- P1.6 and P1.7 bit latches are loaded with logic 1s
- RAM location HADD is loaded with the high-order address byte of the service routines
- The SIO1 interrupt enable and interrupt priority bits are set

- The slave mode is enabled by simultaneously setting the ENS1 and AA bits in S1CON and the serial clock frequency (for master modes) is defined by loading CR0 and CR1 in S1CON. The master routines must be started in the main program.

The SIO1 hardware now begins checking the I<sup>2</sup>C bus for its own slave address and general call. If the general call or the own slave address is detected, an interrupt is requested and S1STA is loaded with the appropriate state information. The following text describes a fast method of branching to the appropriate service routine.

## 80C51 Family Derivatives

## 8XC552/562 overview

## SIO1 INTERRUPT ROUTINE

When the SIO1 interrupt is entered, the PSW is first pushed on the stack. Then S1STA and HADD (loaded with the high-order address byte of the 26 service routines by the initialization routine) are pushed on to the stack. S1STA contains a status code which is the lower byte of one of the 26 service routines. The next instruction is RET, which is the return from subroutine instruction. When this instruction is executed, the high and low order address bytes are popped from stack and loaded into the program counter.

The next instruction to be executed is the first instruction of the state service routine. Seven bytes of program code (which execute in eight machine cycles) are required to branch to one of the 26 state service routines.

SI	PUSH	PSW	Save PSW
	PUSH	S1STA	Push status code (low order address byte)
	PUSH	HADD	Push high order address byte
	RET		Jump to state service routine

The state service routines are located in a 256-byte page of program memory. The location of this page is defined in the initialization routine. The page can be located anywhere in program memory by loading data RAM register HADD with the page number. Page 01 is chosen in this example, and the service routines are located between addresses 0100H and 01FFH.

## THE STATE SERVICE ROUTINES

The state service routines are located 8 bytes from each other. Eight bytes of code are sufficient for most of the service routines. A few of the routines require more than 8 bytes and have to jump to other locations to obtain more bytes of code. Each state routine is part of the SIO1 interrupt routine and handles one of the 26 states. It ends with a RETI instruction which causes a return to the main

program.

## MASTER TRANSMITTER AND MASTER RECEIVER MODES

The master mode is entered in the main program. To enter the master transmitter mode, the main program must first load the internal data RAM with the slave address, data bytes, and the number of data bytes to be transmitted. To enter the master receiver mode, the main program must first load the internal data RAM with the slave address and the number of data bytes to be received. The R/W bit determines whether SIO1 operates in the master transmitter or master receiver mode.

Master mode operation commences when the STA bit in S1CION is set by the SETB instruction and data transfer is controlled by the master state service routines in accordance with Table 3, Table 4, Figure 17, and Figure 18. In the example below, 4 bytes are transferred. There is no repeated START condition. In the event of lost arbitration, the transfer is restarted when the bus becomes free. If a bus error occurs, the I<sup>2</sup>C bus is released and SIO1 enters the not selected slave receiver mode. If a slave device returns a not acknowledge, a STOP condition is generated.

A repeated START condition can be included in the serial transfer if the STA flag is set instead of the STO flag in the state service routines vectored to by status codes 28H and 58H. Additional software must be written to determine which data is transferred after a repeated START condition.

## SLAVE TRANSMITTER AND SLAVE RECEIVER MODES

After initialization, SIO1 continually tests the I<sup>2</sup>C bus and branches to one of the slave state service routines if it detects its own slave address or the general call address (see Table 5, Table 6, Figure 19, and Figure 20). If arbitration was lost while in the master mode, the master mode is restarted after the

current transfer. If a bus error occurs, the I<sup>2</sup>C bus is released and SIO1 enters the not selected slave receiver mode.

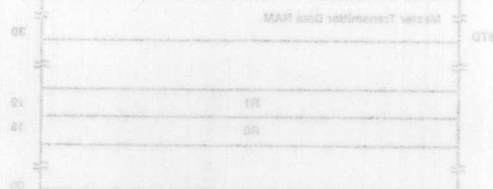
In the slave receiver mode, a maximum of 8 received data bytes can be stored in the internal data RAM. A maximum of 8 bytes ensures that other RAM locations are not overwritten if a master sends more bytes. If more than 8 bytes are transmitted, a not acknowledge is returned, and SIO1 enters the not addressed slave receiver mode. A maximum of one received data byte can be stored in the internal data RAM after a general call address is detected. If more than one byte is transmitted, a not acknowledge is returned and SIO1 enters the not addressed slave receiver mode.

In the slave transmitter mode, data to be transmitted is obtained from the same locations in the internal data RAM that were previously loaded by the main program. After a not acknowledge has been returned by a master receiver device, SIO1 enters the not addressed slave mode.

## ADAPTING THE SOFTWARE FOR DIFFERENT APPLICATIONS

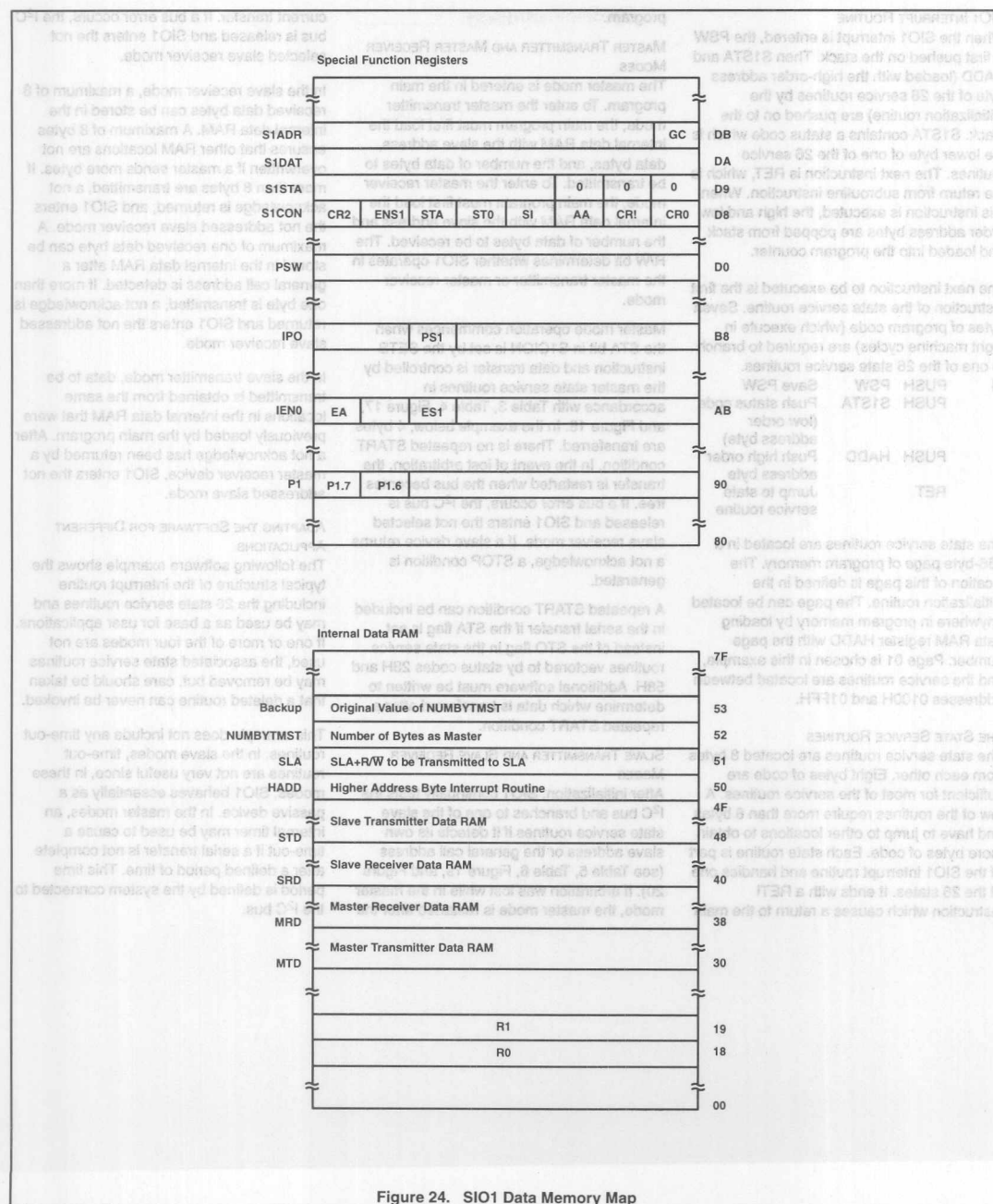
The following software example shows the typical structure of the interrupt routine including the 26 state service routines and may be used as a base for user applications. If one or more of the four modes are not used, the associated state service routines may be removed but, care should be taken that a deleted routine can never be invoked.

This example does not include any time-out routines. In the slave modes, time-out routines are not very useful since, in these modes, SIO1 behaves essentially as a passive device. In the master modes, an internal timer may be used to cause a time-out if a serial transfer is not complete after a defined period of time. This time period is defined by the system connected to the I<sup>2</sup>C bus.



## 80C51 Family Derivatives

## 8XC552/562 overview





## 80C51 Family Derivatives

## 8XC552/562 overview

! SI01 EQUATE LIST			
! LOCATIONS OF THE SI01 SPECIAL FUNCTION REGISTERS			
00D8	S1CON	-0xd8	
00D9	S1STA	-0xd9	
00DA	S1DAT	-0xda	
00DB	S1ADR	-0xdb	
00A8	IEN0	-0xa8	
00B8	IP0	-0xb8	
! BIT LOCATIONS			
00DD	STA	-0xdd	! STA bit in S1CON
00BD	SI01HP	-0xbd	! IP0, SI01 Priority bit
! IMMEDIATE DATA TO WRITE INTO REGISTER S1CON			
00D5	ENS1_NOTSTA_STO_NOTSI_AA_CR0	-0xd5	! Generates STOP ! (CR0 = 100kHz)
00C5	ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0	-0xc5	! Releases BUS and ! ACK
00C1	ENS1_NOTSTA_NOTSTO_NOTSI_NOTAA_CR0	-0xc1	! Releases BUS and ! NOT ACK
00E5	ENS1_STA_NOTSTO_NOTSI_AA_CR0	-0xe5	! Releases BUS and ! set STA
! GENERAL IMMEDIATE DATA			
0031	OWNSLA	-0x31	! Own SLA+General Call ! must be written into S1ADR
00A0	ENSI01	-0xa0	! EA+ES1, enable SI01 interrupt ! must be written into IEN0
0001	PAG1	-0x01	! select PAG1 as HADD
00C0	SLAW	-0xc0	! SLA+W to be transmitted
00C1	SLAR	-0xc1	! SLA+R to be transmitted
0018	SELRB3	-0x18	! Select Register Bank 3
! LOCATIONS IN DATA RAM			
0030	MTD	-0x30	! MST/TRX/DATA base address
0038	MRD	-0x38	! MST/REC/DATA base address
0040	SRD	-0x40	! SLV/REC/DATA base address
0048	STD	-0x48	! SLV/TRX/DATA base address
0053	BACKUP	-0x53	! Backup from NUMBYTMST ! To restore NUMBYTMST in case ! of an Arbitration Loss.
0052	NUMBYTMST	-0x52	! Number of bytes to transmit ! or receive as MST.
0051	SLA	-0x51	! Contains SLA+R/W to be ! transmitted.
0050	HADD	-0x50	! High Address byte for STATE 0 ! till STATE 25.

## 80C51 Family Derivatives

## 8XC552/562 overview

*****			
! INITIALIZATION ROUTINE			
! Example to initialize IIC Interface as slave receiver or			
! slave transmitter and start a MASTER TRANSMIT or a MASTER			
! RECEIVE function. 4 bytes will be transmitted or received.			
*****			
	.sect strt		
	.base 0x00		
0000	4100	ajmp INIT	! RESET
*****			
	.sect initial		
	.base 0x200		
0200	75DB31	INIT:	
		mov S1ADR,#OWNSLA	! Load own SLA + enable
			! general call recognition
0203	D296	setb P1(6)	! P1.6 High level.
0205	D297	setb P1(7)	! P1.7 High level.
0207	755001	mov HADD,#PAG1	
020A	43A8A0	orl IEN0,#ENSI01	! Enable SI01 interrupt
020D	C2BD	clr SI01HP	! SI01 interrupt low
			! priority
020F	75D8C5	mov S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0	! Initialize SLV funct.
*****			
! IMMEDIATE DATA TO WRITE INTO REGISTER S1CON			
-----			
! START MASTER TRANSMIT FUNCTION			
-----			
0212	755204	mov NUMBYTMST,#0x4	! Transmit 4 bytes.
0215	7551C0	mov SLA,#SLAW	! SLA+W, Transmit funct.
0218	D2DD	setb STA	! set STA in S1CON
-----			
! START MASTER RECEIVE FUNCTION			
-----			
021A	755204	mov NUMBYTMST,#0x4	! Receive 4 bytes.
021D	7551C1	mov SLA,#SLAR	! SLA+R, Receive funct.
0220	D2DD	setb STA	! set STA in S1CON
*****			
! LOCATION IN DATA RAM			
-----			
		STC	0000
		SRD	0000
		MRD	0000
		MTC	0000
		BACKUP	0000
		NUMBYTMST	0000
		SLA	0000
		HADD	0000
*****			

## 80C51 Family Derivatives

## 8XC552/562 overview

		-----	
		! S101 INTERRUPT ROUTINE	
		-----	
		.sect intvec	! S101 interrupt vector
		base 0x00	
		! S1STA and HADD are pushed onto the stack. They serve as	
		! return address for the RET instruction.	
		! The RET instruction sets the Program Counter to address	
		! HADD, S1STA and jumps to the right subroutine.	
002B	C0D0	push psw	! save psw
002D	C0D9	push S1STA	
002F	C050	push HADD	
0031	22	ret	! JMP to address
		! HADD,S1STA.	
		-----	
		! STATE : 00, Bus error.	
		! ACTION: Enter not addressed SLV mode and release bus.	
		STO reset.	
		-----	
		.sect st0	
		base 0x100	
0100	75D8D5	mov S1CON,#ENS1_NOTSTA_STO_NOTSI_AA_CR0	! clr SI
		! set STO,AA	
0103	D0D0	pop psw	
0105	32	reti	

## 80C51 Family Derivatives

## 8XC552/562 overview

```

*****
*****
! MASTER STATE SERVICE ROUTINES
*****
! State 08 and State 10 are both for MST/TRX and MST/REC.
! The R/W bit decides whether the next state is within
! MST/TRX mode or within MST/REC mode.
*****
! The RET instruction sets the Program Counter to address
! HADD_21STA and jumps to the next subroutine.
-----
! STATE : 08, A, START condition has been transmitted.
! ACTION: SLA+R/W are transmitted, ACK bit is received.
-----
.sect mts8
.base 0x108

0108 8551DA mov S1DAT,SLA ! Load SLA+R/W
010B 75D8C5 mov S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
! ACTION: Enter not addressed SLV mode and release bus
! clr SI
010E 01A0 ajmp INITBASE1
-----
! STATE : 10, A repeated START condition has been
! transmitted.
! ACTION: SLA+R/W are transmitted, ACK bit is received.
-----
.sect mts10
.base 0x110

0110 8551DA mov S1DAT,SLA ! Load SLA+R/W
0113 75D8C5 mov S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
! clr SI
010E 01A0 ajmp INITBASE1

.sect ibase1
.base 0xa0
INITBASE1:
00A0 75D018 mov psw,#SELRB3
00A3 7930 mov r1,#MTD
00A5 7838 mov r0,#MRD
00A7 855253 mov BACKUP,NUMBYTMST ! Save initial value
00AA D0D0 pop psw
00AC 32 reti

```

## 80C51 Family Derivatives

## 8XC552/562 overview

```

-----
!
! *****
! MASTER TRANSMITTER STATE SERVICE ROUTINES
! *****
!
!-----
!
!-----
! STATE : 18, Previous state was STATE 8 or STATE 10,
! SLA+W have been transmitted, ACK has
! been received.
! ACTION: First DATA is transmitted, ACK bit
! is received.
!-----
.sect mts18
.base 0x118
! STATE : 18, Address of DATA
! ACTION: Bus is released, not addressed SLV mode
! If a new START condition is transmitted
! the IIC bus is free again.
0118 75D018 mov psw,#SELRB3
011B 87DA mov S1DAT,@r1
011D 01B5 ajmp CON
!-----
! STATE : 20, SLA+W have been transmitted, NOT ACK
! has been received
! ACTION: Transmit STOP condition.
!-----
.sect mts20
.base 0x110
0120 75D8D5 mov S1CON,#ENS1_NOTSTA_STO_NOTSI_AA_CR0
! set STO, clr SI
0123 D0D0 pop psw
0125 32 reti
!-----
! STATE : 28, DATA of S1DAT have been transmitted,
! ACK received.
! ACTION: If Transmitted DATA is last DATA then
! transmit a STOP condition, else transmit
! next DATA.
!-----
.sect mts28
.base 0x128
0128 D55285 mov d1n NUMBYTMST,NOTLDAT1 ! JMP if NOT last DATA
012B 75D8D5 mov S1CON,#ENS1_NOTSTA_STO_NOTSI_AA_CR0
! set STO, clr SI
012E 01B9 ajmp RETmt
!-----
.sect mts28sb
.base 0x0b0
00B0 75D018 NOTLDAT1: mov psw,#SELRB3
00B3 87DA mov S1DAT,@r1
00B5 75D8C5 CON: mov S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
! clr SI, set AA
00B8 09 inc r1
00B9 D0D0 RETmt : pop psw
00BB 32 reti
!-----

```



## 80C51 Family Derivatives

## 8XC552/562 overview

```

-----
! STATE : 30, DATA of S1DAT have been transmitted,
! ACTION: NOT ACK received.
! ACTION: Transmit a STOP condition.
-----
.sect mts30
.base 0x130
0130 75D8D5 mov S1CON,#ENS1_NOTSTA_STO_NOTSI_AA_CR0 ! set STO, clr SI
0133 D0D0 pop psw
0135 32 reti

-----
! STATE : 38, Arbitration lost in SLA+W or DATA.
! ACTION: Bus is released, not addressed SLV mode is
entered. A new START condition is transmitted
when the IIC bus is free again.
-----
.sect mts38
.base 0x138
0138 75D8E5 mov S1CON,#ENS1_STA_NOTSTO_NOTSI_AA_CR0
013B 855352 mov NUMBYTMST,BACKUP
013E 01B9 ajmp RETmt

.sect mts30
.base 0x130
0130 75D8D5 mov S1CON,#ENS1_NOTSTA_STO_NOTSI_AA_CR0 ! set STO, clr SI

-----
! MASTER RECEIVER STATE SERVICE ROUTINES
-----
! STATE : 40, Previous state was STATE 08 or STATE 10,
SLA+R have been transmitted, ACK received.
! ACTION: DATA will be received, ACK returned.
-----
.sect mts40
.base 0x140
0140 75D8C5 mov S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0 ! clr STA, STO, SI set AA
0143 01CB pop psw
reti

-----
! STATE : 48, SLA+R have been transmitted, NOT ACK
received.
! ACTION: STOP condition will be generated.
-----
.sect mts48
.base 0x148
0148 75D8D5 mov S1CON,#ENS1_NOTSTA_STO_NOTSI_AA_CR0 ! set STO, clr SI
014B D0D0 pop psw
014D 32 reti

```

## 80C51 Family Derivatives

## 8XC552/562 overview

		-----	
		! STATE : 50, DATA have been received, ACK returned.	
		! ACTION : Read DATA of S1DAT.	
		! DATA will be received, if it is last	
		! DATA then NOT ACK will be returned else ACK	
		! will be returned.	
		-----	
		.sect mrs50	
		.base 0x150	
0150	75D018	mov	psw,#SELRB3
0153	A6DA	mov	@r0,S1DAT ! Read received DATA
0155	01C0	ajmp	REC1
		-----	
		.sect mrs50s	
		.base 0xc0	
00C0	D55205	REC1:	djnz NUMBYTMST,NOTLDAT2
00C3	75D8C1	mov	S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_NOTAA_CR0
		! clr SI,AA	
00C6	8003	sjmp	RETmr
00C8	75D8C5	NOTLDAT2:	mov S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
		! clr SI, set AA	
00CB	08	RETmr:	inc r0
00CC	D0D0	pop	psw
00CE	32	reti	
		-----	
		! STATE : 58, DATA have been received, NOT ACK	
		! returned.	
		! ACTION : Read DATA of S1DAT and generate a STOP	
		! condition.	
		-----	
		.sect mrs58	
		.base 0x158	
0158	75D018	mov	psw,#SELRB3
015B	A6DA	mov	@r0,S1DAT
015D	80E9	sjmp	STOP
		-----	
		! STATE : 70, General call has been received, ACK	
		! returned.	
		! ACTION : DATA will be received and ACK returned.	
		! STA is set to reset M8T mode after the	
		! bus is free again.	
		-----	
		.sect mrs70	
		.base 0x170	
0170	75D8C5	mov	S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
		! clr SI, set AA	
0173	75D018	mov	psw,#SELRB3
		! initialize S1RD counter	
		-----	
		! STATE : 7B, Arbitration lost in S1A-RW as M8T.	
		! General call has been received, ACK returned.	
		! ACTION : DATA will be received and ACK returned.	
		! STA is set to reset M8T mode after the	
		! bus is free again.	
		-----	
		.sect mrs7B	
		.base 0x17B	
017B	75D8C5	mov	S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
		! initialize S1RD counter	
017E	01D0	ajmp	INITRD

## 80C51 Family Derivatives

## 8XC552/562 overview

```

-----
! SLAVE RECEIVER STATE SERVICE ROUTINES
-----
! STATE : 60, Own SLA+W have been received, ACK
!         returned.
! ACTION: DATA will be received and ACK returned.
-----
.sect srs60
.base 0x160
0160 75D8C5      mov  S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
0163 75D018      mov  psw,#SELRB3
0166 01D0        ajmp INITSRD
        .sect insrd
        .base 0xd0
00D0 7840        mov  r0,#SRD
00D2 7908        mov  r1,#8
00D4 D0D0        pop  psw
00D6 32          reti

-----
! STATE : 68, Arbitration lost in SLA and R/W as MST
!         Own SLA+W have been received, ACK returned
! ACTION: DATA will be received and ACK returned.
!         STA is set to restart MST mode after the
!         bus is free again.
-----
.sect srs68
.base 0x168
0168 75D8E5      mov  S1CON,#ENS1_STA_NOTSTO_NOTSI_AA_CR0
016B 75D018      mov  psw,#SELRB3
016E 01D0        ajmp INITSRD

-----
! STATE : 70, General call has been received, ACK
!         returned.
! ACTION: DATA will be received and ACK returned.
-----
.sect srs70
.base 0x170
0170 75D8C5      mov  S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
0173 75D018      mov  psw,#SELRB3
        ! Initialize SRD counter

-----
! STATE : 78, Arbitration lost in SLA+R/W as MST.
!         General call has been received, ACK returned.
! ACTION: DATA will be received and ACK returned.
!         STA is set to restart MST mode after the
!         bus is free again.
-----
.sect srs78
.base 0x178
0178 75D8E5      mov  S1CON,#ENS1_STA_NOTSTO_NOTSI_AA_CR0
017B 75D018      mov  psw,#SELRB3
017E 01D0        ajmp INITSRD
        ! Initialize SRD counter

```

## 80C51 Family Derivatives

## 8XC552/562 overview

```

-----
! STATE : 80, Previously addressed with own SLA.
! DATA received, ACK returned.
! ACTION: Read DATA.
! IF received DATA was the last THEN superfluous
! DATA will be received and NOT ACK returned ELSE
! next DATA will be received and ACK returned.
-----
.sect srs80
.base 0x180

0180 75D018      mov     psw,#SELRB3
0183 A6DA      mov     @r0,S1DAT      ! Read received DATA
0185 01D8      ajmp    REC2

.sect srs80s
.base 0xd8

00D8 D906      REC2: djnz   r1,NOTLDAT3
00DA 75D8C1      mov     S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_NOTAA_CR0
                                ! clr SI,AA
00DD D0D0      pop     psw
00DF 32         reti
00E0 75D8C5      NOTLDAT2: mov   S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                ! clr SI, set AA
00E3 08         inc     r0
00E4 D0D0      RETsr:  pop    psw
00E6 32         reti

-----
! STATE : 88, Previously addressed with own SLA.
! DATA received NOT ACK returned.
! ACTION: No save of DATA, Enter NOT addressed SLV
! mode. Recognition of own SLA. General call
! recognized, if S1ADR. 0-1.
-----
.sect srs88
.base 0x188

0188 75D8C5      mov     S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                ! clr SI, set AA
018B 01E4      ajmp    RETsr

-----
! STATE : 90, Previously addressed with general call.
! DATA has been received, ACK has been returned.
! ACTION: Read DATA.
! After General call only one byte will be
! received with ACK the second DATA will be
! received with NOT ACK.
! DATA will be received and NOT ACK
! returned.
-----
.sect srs90
.base 0x190

0190 75D018      mov     psw,#SELRB3
0193 A6DA      mov     @r0,S1DAT      ! Read received DATA
0195 01DA      ajmp    LDAT

```

80C51 Family Derivatives

8XC552/562 overview

		STATE :		98, Previously addressed with general call. DATA has been received, NOT ACK has been returned.			
		ACTION :		No save of DATA, Enter NOT addressed SLV mode. Recognition of own SLA. General call recognized, if S1ADR. 0-1.			
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
0198	75D8C5	mov	S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0	mov	S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0	! clr SI, set AA	
019B	D0D0	pop	psw	pop	psw		
019D	32	reti		reti			
		STATE :		A0, A STOP condition or repeated START has been received, while still addressed as SLV/REC or SLV/TRX.			
		ACTION :		No save of DATA, Enter NOT addressed SLV mode. Recognition of own SLA. General call recognized, if S1ADR. 0-1.			
		.sect srsA0		.sect srsA0			
		.base 0x1A0		.base 0x1A0			
01A0	75D8C5	mov	S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0	mov	S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0	! clr SI, set AA	
01A3	D0D0	pop	psw	pop	psw		
01A5	32	reti		reti			
		STATE :		98, Previously addressed with general call. DATA has been received, NOT ACK has been returned.			
		ACTION :		No save of DATA, Enter NOT addressed SLV mode. Recognition of own SLA. General call recognized, if S1ADR. 0-1.			
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			
		.base 0x198		.base 0x198			
		mov		S1CON,#ENS1_NOTSTA_NOTSTO_NOTSL_AA_CR0		! clr SI, set AA	
		.sect srs98		.sect srs98			



## 80C51 Family Derivatives

## 8XC552/562 overview

*****			
! SLAVE TRANSMITTER STATE SERVICE ROUTINES			
*****			
-----			
		! STATE :	A8, Own SLA+R received, ACK returned.
		! ACTION:	DATA will be transmitted, A bit received.
-----			
		.sect stsa8	
		.base 0x1a8	
-----			
01A8	8548DA	mov	S1DAT,STD ! load DATA in S1DAT
01AB	75D8C5	mov	S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CRO
			! ACTION:
			! clr SI, set AA
01AE	01E8	ajmp	INITBASE2
-----			
		.sect ibase2	
		.base 0xe8	
00E8	75D018	mov	psw,#SELRB3
00EB	7948	mov	r1,#STD
00ED	09	inc	r1
00EE	D0D0	pop	psw
00F0	32	reti	
-----			
! STATE : B0, Arbitration lost in SLA and R/W as MST.			
! Own SLA+R received, ACK returned.			
! ACTION: DATA will be transmitted, A bit received.			
! STA is set to restart MST mode after the			
! bus is free again.			
! -----			
		.sect stsb0	
		.base 0x1b0	
01B0	8548DA	mov	S1DAT,STD ! load DATA in S1DAT
01B3	75D8E5	mov	S1CON,#ENS1_STA_NOTSTO_NOTSI_AA_CRO
01B6	01E8	ajmp	INITBASE2
! STATE : B8, DATA has been transmitted, ACK received,			
! ACTION: DATA will be transmitted, ACK bit is received.			
! -----			
		.sect stsb8	
		.base 0x1b8	
01B8	75D018	mov	psw,#SELRB3
01BB	87DA	mov	S1DAT,@r1
01BD	01F8	ajmp	SCON
-----			
		.sect scn	
		.base 0xf8	
00F8	75D8C5	SCON:	mov S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CRO
			! clr SI, set AA
00FB	09	inc	r1
00FC	D0D0	pop	psw
00FE	32	reti	

80C51 Family Derivatives

8XC552/562 overview

```
-----
! STATE : C0, DATA has been transmitted, NOT ACK
! ACTION: Enter not addressed SLV mode.
-----
.sect stsc0
.base 0x1c0
01C0 75D8C5 mov S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CRO
01C3 D0D0 ! clr SI, set AA
01C5 32 pop psw
reti
-----
! STATE : C8, Last DATA has been transmitted (AA=0),
! ACTION: ACK received.
! ACTION: Enter not addressed SLV mode.
-----
.sect stsc8
.base 0x1c8
01C8 75D8C5 mov S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CRO
01CB D0D0 ! clr SI, set AA
01CD 32 pop psw
reti
-----
! END OF SI01 INTERRUPT ROUTINE
-----
```

Reset Circuitry

The reset circuitry for the 8XC552 is connected to the reset pin RST. A Schmitt trigger is used at the input for noise rejection (see Figure 25). The output of the Schmitt trigger is sampled by the reset circuitry every machine cycle.

A reset is accomplished by holding the RST pin HIGH for at least two machine cycles (24 oscillator periods) while the oscillator is running. The CPU responds by executing an internal reset. During reset, ALE and PSEN output a HIGH level. In order to perform a correct reset, this level must not be affected by external elements. The RST line can also be pulled HIGH internally by a pull-up transistor activated by the watchdog timer T3. The length of the output pulse from T3 is 3 machine cycles. A pulse of such short duration is necessary in order to recover from a processor or system fault as fast as possible.

Note that the short reset pulse from Timer T3 cannot discharge the power-on reset capacitor (see Figure 26). Consequently, when the watchdog timer is also used to set external devices, this capacitor arrangement should not be connected to the RST pin, and a different circuit should be used to perform the power-on reset operation. A timer T3

overflow, if enabled, will force a reset condition to the 8XC552 by an internal connection, whether the output RST is tied LOW or not.

The internal reset is executed during the second cycle in which RST is HIGH and is repeated every cycle until RST goes low. It leaves the internal registers as follows:

## 80C51 Family Derivatives

## 8XC552/562 overview

REGISTER	CONTENT	
ACC	0000	0000
ADCON	xx00	0000
ADCH	xxxx	xxxx
B	0000	0000
CML0-CML2	0000	0000
CMH0-CMH2	0000	0000
CTCON	0000	0000
CTL0-CTL3	xxxx	xxxx
CTH0-CTH3	xxxx	xxxx
DPL	0000	0000
DPH	0000	0000
IEN0	0000	0000
IEN1	0000	0000
IP0	x000	0000
IP1	0000	0000
PCH	0000	0000
PCL	0000	0000
PCON	0xx0	0000
PSW	0000	0000
PWM0	0000	0000
PWM1	0000	0000
PWMP	0000	0000
P0-P4	1111	1111
P5	xxxx	xxxx
RTE	0000	0000
S0BUF	xxxx	xxxx
S0CON	0000	0000
S1ADR	0000	0000
S1CON	0000	0000
S1DAT	0000	0000
S1STA	1111	1000
SP	0000	0111
STE	1100	0000
TCON	0000	0000
TH0, TH1	0000	0000
TMH2	0000	0000
TL0, TL1	0000	0000
TML2	0000	0000
TMOD	0000	0000
TM2CON	0000	0000
TM2IR	0000	0000
T3	0000	0000

The internal RAM is not affected by reset. At power-on, the RAM content is indeterminate.

### Interrupts

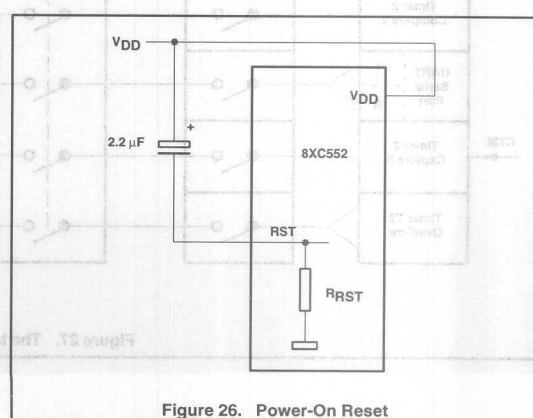
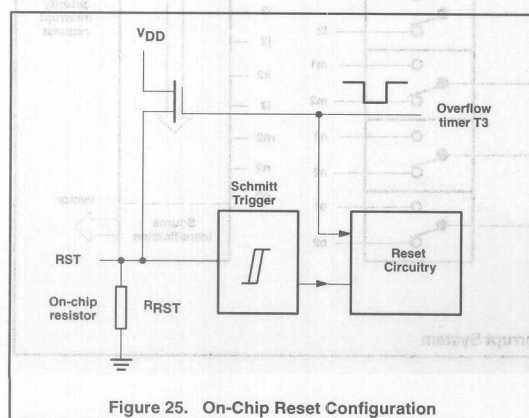
The 8XC552 has fifteen interrupt sources, each of which can be assigned one of two priority levels, as shown in Figure 27. The five interrupt sources common to the 80C51 are the external interrupts (INT0 and INT1), the timer 0 and timer 1 interrupts (IT0 and IT1), and the serial I/O interrupt (RI or TI). In the 8XC552, the standard serial interrupt is called SIO0. Since the subsystems which create these interrupts are identical on both parts, their functionality is likewise identical. The only differences are the locations of the enable and priority register configurations and the priority structure. This is detailed below along with the specifics of the interrupts unique to the 8XC552.

The eight Timer T2 interrupts are generated by flags CT10-CT13, CM10-CM12, and by the logical OR of flags T2OV and T2BO. Flags CT10 to CT13 are set by input signals CT0I to CT3I. Flags CM10 to CM12 are set when a match occurs between Timer T2 and the

compare registers CM0, CM1, and CM2.

When an 8-bit or 16-bit overflow occurs, flags T2BO and T2OV are set, respectively. These nine flags are not cleared by hardware and must be reset by software to avoid recurring interrupts.

The ADC interrupt is generated by the ADCl flag in the ADC control register (ADCON). This flag is set when an ADC conversion result is ready to be read. ADCl is not cleared by hardware and must be reset by software to avoid recurring interrupts.



## 80C51 Family Derivatives

## 8XC552/562 overview

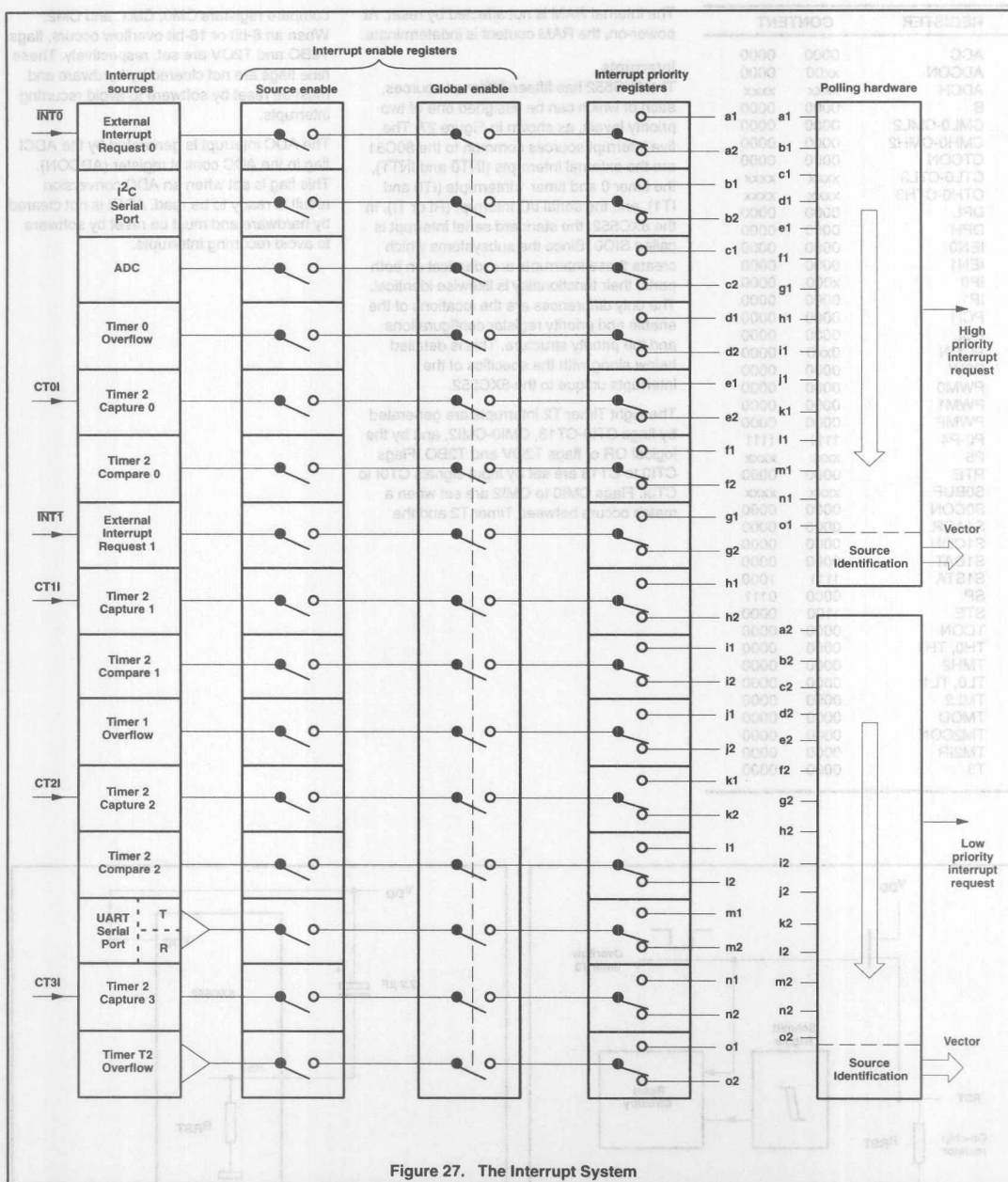


Figure 27. The Interrupt System

## 80C51 Family Derivatives

## 8XC552/562 overview

The SIO1 (I<sup>2</sup>C) interrupt is generated by the SI flag in the SIO1 control register (S1CON). This flag is set when S1STA is loaded with a valid status code.

The ADC1 flag may be reset by software. It cannot be set by software. All other flags that generate interrupts may be set or cleared by software, and the effect is the same as setting or resetting the flags by hardware. Thus, interrupts may be generated by software and pending interrupts can be canceled by software.

**Interrupt Enable Registers:** Each interrupt source can be individually enabled or disabled by setting or clearing a bit in the interrupt enable special function registers IEN0 and IEN1. All interrupt sources can also be globally enabled or disabled by setting or clearing bit EA in IEN0. The interrupt enable registers are described in Figures 28 and 29.

**Interrupt Priority Structure:** Each interrupt source can be assigned one of two priority levels. Interrupt priority levels are defined by the interrupt priority special function registers IPO and IP1. IPO and IP1 are described in Figures 30 and 31.

Interrupt priority levels are as follows:

"0"—low priority  
"1"—high priority

A low priority interrupt may be interrupted by a high priority interrupt. A high priority interrupt cannot be interrupted by any other interrupt source. If two requests of different priority occur simultaneously, the high priority

level request is serviced. If requests of the same priority are received simultaneously, an internal polling sequence determines which request is serviced. Thus, within each priority level, there is a second priority structure determined by the polling sequence. This second priority structure is shown in Table 8.

The above Priority Within Level structure is only used when there are simultaneous requests of the same priority level.

**Interrupt Handling:** The interrupt sources are sampled at S5P2 of every machine cycle. The samples are polled during the following machine cycle. If one of the flags was in a set condition at S5P2 of the previous machine cycle, the polling cycle will find it and the interrupt system will generate an LCALL to the appropriate service routine, provided this hardware-generated LCALL is not blocked by any of the following conditions:

1. An interrupt of higher or equal priority level is already in progress.
2. The current machine cycle is not the final cycle in the execution of the instruction in progress. (No interrupt request will be serviced until the instruction in progress is completed.)
3. The instruction in progress is RETI or any access to the interrupt priority or interrupt enable registers. (No interrupt will be serviced after RETI or after a read or write to IPO, IP1, IEO, or IE1 until at least one other instruction has been subsequently executed.)

The polling cycle is repeated with every machine cycle, and the values polled are the values present at S5P2 of the previous machine cycle. Note that if an interrupt flag is active but is not being responded to because of one of the above conditions, and if the flag is inactive when the blocking condition is removed, then the blocked interrupt will not be serviced. Thus, the fact that the interrupt flag was once active but not serviced is not remembered. Every polling cycle is new.

The processor acknowledges an interrupt request by executing a hardware-generated LCALL to the appropriate service routine. In some cases it also clears the flag which generated the interrupt, and in others it does not. It clears the Timer 0, Timer 1, and external interrupt flags. An external interrupt flag (IE0 or IE1) is cleared only if it was transition-activated. All other interrupt flags are not cleared by hardware and must be cleared by the software. The LCALL pushes the contents of the program counter on to the stack (but it does not save the PSW) and reloads the PC with an address that depends on the source of the interrupt being vectored to as shown in Table 9.

Execution proceeds from the vector address until the RETI instruction is encountered. The RETI instruction clears the "priority level active" flip-flop that was set when this interrupt was acknowledged. It then pops the top two bytes from the stack and reloads the program counter. Execution of the interrupted program continues from where it was interrupted.

		7	6	5	4	3	2	1	0
IEN0 (A8H)		EA	EAD	ES1	ES0	ET1	EX1	ET0	EX0
		(MSB)				(LSB)			
Bit	Symbol	Function							
IEN0.7	EA	Global enable/disable control 0 = No interrupt is enabled 1 = Any individually enabled interrupt will be accepted							
IEN0.6	EAD	Enable ADC interrupt							
IEN0.5	ES1	Enable SIO1 (I <sup>2</sup> C) interrupt							
IEN0.4	ES0	Enable SIO0 (UART) interrupt							
IEN0.3	ET1	Enable Timer 1 interrupt							
IEN0.2	EX1	Enable External interrupt 1							
IEN0.1	ET0	Enable Timer 0 interrupt							
IEN0.0	EX0	Enable External interrupt 0							

**Figure 28. Interrupt Enable Register (IEN0)**

Figure 28. Interrupt Enable Register (IEN0)

		7	6	5	4	3	2	1	0
IEN1 (E8H)		ET2	ECM2	ECM1	ECM0	ECT3	ECT2	ECT1	ECT0
	(MSB)								(LSB)

Bit	Symbol	Function
IEN1.7	ET2	Enable T2 overflow interrupt(s)
IEN1.6	ECM2	Enable T2 comparator 2 interrupt
IEN1.5	ECM1	Enable T2 comparator 1 interrupt
IEN1.4	ECM0	Enable T2 comparator 0 interrupt
IEN1.3	ECT3	Enable T2 capture register 3 interrupt
IEN1.2	ECT2	Enable T2 capture register 2 interrupt
IEN1.1	ECT1	Enable T2 capture register 1 interrupt
IEN1.0	ECT0	Enable T2 capture register 0 interrupt

In all cases, if the enable bit is 0, then the interrupt is disabled, and if the enable bit is 1, then the interrupt is enabled.

**Figure 29. Interrupt Enable Register (IEN1)**

Figure 29. Interrupt Enable Register (IEN1)



80C51 Family Derivatives

8XC552/562 overview

	7	6	5	4	3	2	1	0
IP0 (B8H)	<div><div></div><div></div></div>	PAD	PS1	PS0	PT1	PX1	PT0	PX0
	(MSB)				(LSB)			
Bit	Symbol		Function					
IP.7	-		Unused					
IP.6	PAD		ADC Interrupt priority level					
IP.5	PS1		SIO1 (I <sup>2</sup> C) interrupt priority level					
IP.4	PS0		SIO0 (UART) interrupt priority level					
IP.3	PT1		Timer 1 interrupt priority level					
IP.2	PX1		External Interrupt 1 priority level					
IP.1	PT0		Timer 0 interrupt priority level					
IP.0	PX0		External Interrupt 0 priority level					

**Figure 30. Interrupt Priority Register (IP0)**

Figure 30. Interrupt Priority Register (IP0)

7	6	5	4	3	2	1	0	
IP1 (F8H)	PT2	PCM2	PCM1	PCM0	PCT3	PCT2	PCT1	PCT0
(MSB)				(LSB)				
Bit	Symbol	Function						
IP1.7	PT2	T2 overflow interrupt(s) priority level						
IP1.6	PCM2	T2 comparator 2 interrupt priority level						
IP1.5	PCM1	T2 comparator 1 interrupt priority level						
IP1.4	PCM0	T2 comparator 0 interrupt priority level						
IP1.3	PCT3	T2 capture register 3 interrupt priority level						
IP1.2	PCT2	T2 capture register 2 interrupt priority level						
IP1.1	PCT1	T2 capture register 1 interrupt priority level						
IP1.0	PCT0	T2 capture register 0 interrupt priority level						

**Figure 31. Interrupt Priority Register (IP1)**

Figure 31. Interrupt Priority Register (IP1)

Table 8. Interrupt Priority Structure

SOURCE	NAME	PRIORITY WITHIN LEVEL
External interrupt 0	X0	(highest)
SIO1 (I <sup>2</sup> C)	S1	↑
ADC completion	ADC	
Timer 0 overflow	T0	
T2 capture 0	CT0	
T2 compare 0	CM0	
External interrupt 1	X1	
T2 capture 1	CT1	
T2 compare 1	CM1	
Timer 1 overflow	T1	
T2 capture 2	CT2	
T2 compare 2	CM2	
SIO0 (UART)	S0	
T2 capture 3	CT3	
Timer T2 overflow	T2	(lowest)

Table 9. Interrupt Vector Addresses

SOURCE	NAME	VECTOR ADDRESS
External interrupt 0	X0	0003H
Timer 0 overflow	T0	000BH
External interrupt 1	X1	0013H
Timer 1 overflow	T1	001BH
SIO0 (UART)	S0	0023H
SIO1 (I <sup>2</sup> C)	S1	002BH
T2 capture 0	CT0	0033H
T2 capture 1	CT1	003BH
T2 capture 2	CT2	0043H
T2 capture 3	CT3	004BH
ADC completion	ADC	0053H
T2 compare 0	CM0	005BH
T2 compare 1	CM1	0063H
T2 compare 2	CM2	006BH
T2 overflow	T2	0073H

## 80C51 Family Derivatives

## 8XC552/562 overview

## I/O Port Structure

The 8XC552 has six 8-bit ports. Each port consists of a latch (special function registers P0 to P5), an input buffer, and an output driver (port 0 to 4 only). Ports 0-3 are the same as in the 80C51, with the exception of the additional functions of port 1. The parallel I/O function of port 4 is equal to that of ports 1, 2, and 3. Port 5 may be used as an input port only.

Figure 32 shows the bit latch and I/O buffer functional diagrams of the unique 8XC552 ports. A bit latch corresponds to one bit in a port's SFR and is represented as a D type flip-flop. A "write to latch" signal from the CPU latches a bit from the internal bus and a "read latch" signal from the CPU places the Q output of the flip-flop on the internal bus. A "read pin" signal from the CPU places the actual port pin level on the internal bus. Some instructions that read a port read the actual port pin levels, and other instructions read the latch (SFR) contents.

## Port 1 Operation

Port 1 operates the same as it does in the 8051 with the exception of port lines P1.6 and P1.7, which may be selected as the SCL and SDA lines of serial port SIO1 (I<sup>2</sup>C). Because the I<sup>2</sup>C bus may be active while the device is disconnected from V<sub>DD</sub>, these pins are provided with open drain drivers. Therefore pins P1.6 and P1.7 do not have internal pull-ups.

## Port 5 Operation

Port 5 may be used to input up to 8 analog signals to the ADC. Unused ADC inputs may be used to input digital inputs. These inputs have an inherent hysteresis to prevent the input logic from drawing excessive current from the power lines when driven by analog signals. Channel to channel crosstalk (Ct) should be taken into consideration when both analog and digital signals are simultaneously input to Port 5 (see, D.C. characteristics in data sheet).

Port 5 is not bidirectional and may not be configured as an output port. All six ports are multifunctional, and their alternate functions are listed in Table 10. A more detailed description of these features can be found in the relevant parts of this section.

## Pulse Width Modulated Outputs

The 8XC552 contains two pulse width modulated output channels (see Figure 33). These channels generate pulses of programmable length and interval. The repetition frequency is defined by an 8-bit prescaler PWMP, which supplies the clock for the counter. The prescaler and counter are common to both PWM channels. The 8-bit counter counts modulo 255, i.e., from 0 to 254 inclusive. The value of the 8-bit counter is compared to the contents of two registers: PWM0 and PWM1. Provided the contents of either of these registers is greater than the counter value, the corresponding PWM0 or PWM1 output is set LOW. If the contents of these registers are equal to, or less than the counter value, the output will be HIGH. The pulse-width-ratio is therefore defined by the contents of the registers PWM0 and PWM1. The pulse-width-ratio is in the range of 0 to 1 and may be programmed in increments of 1/255.

Buffered PWM outputs may be used to drive DC motors. The rotation speed of the motor would be proportional to the contents of PWMn. The PWM outputs may also be configured as a dual DAC. In this application, the PWM outputs must be integrated using conventional operational amplifier circuitry. If the resulting output voltages have to be accurate, external buffers with their own analog supply should be used to buffer the PWM outputs before they are integrated. The repetition frequency  $f_{PWM}$ , at the PWMn outputs is given by:

$$f_{PWM} = \frac{f_{osc}}{2 \times (1 + PWMP) \times 255}$$

This gives a repetition frequency range of 123Hz to 31.4kHz ( $f_{osc} = 16\text{MHz}$ ). At

$f_{osc} = 24\text{MHz}$ , the frequency range is 184Hz to 47.1Hz. By loading the PWM registers with either 00H or FFH, the PWM channels will output a constant HIGH or LOW level, respectively. Since the 8-bit counter counts modulo 255, it can never actually reach the value of the PWM registers when they are loaded with FFH.

When a compare register (PWM0 or PWM1) is loaded with a new value, the associated output is updated immediately. It does not have to wait until the end of the current counter period. Both PWMn output pins are driven by push-pull drivers. These pins are not used for any other purpose.

Prescaler frequency control register PWMP

PWMP (FEH)	7	6	5	4	3	2	1	0
	MSB							LSB

PWMP.0-7 Prescaler division factor = PWMP + 1.

Reading PWMP gives the current reload value. The actual count of the prescaler cannot be read.

PWM0 (FCH) PWM1 (FDH)	7	6	5	4	3	2	1	0
	MSB							LSB

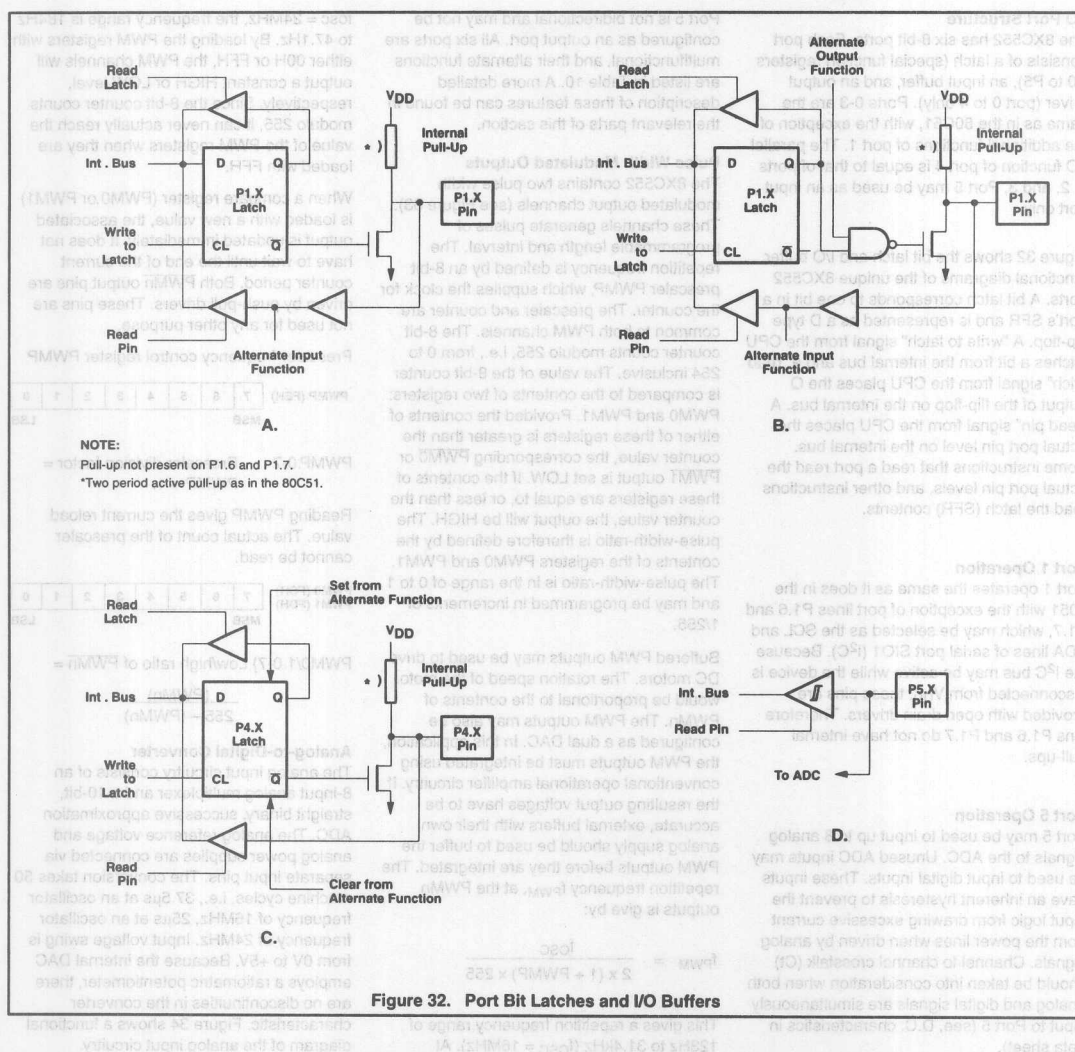
PWM0/1.0-7 Low/high ratio of PWMn =  $\frac{(PWMn)}{255 - (PWMn)}$

## Analog-to-Digital Converter

The analog input circuitry consists of an 8-input analog multiplexer and a 10-bit, straight binary, successive approximation ADC. The analog reference voltage and analog power supplies are connected via separate input pins. The conversion takes 50 machine cycles, i.e., 37.5μs at an oscillator frequency of 16MHz, 25μs at an oscillator frequency of 24MHz. Input voltage swing is from 0V to +5V. Because the internal DAC employs a ratiometric potentiometer, there are no discontinuities in the converter characteristic. Figure 34 shows a functional diagram of the analog input circuitry.

## 80C51 Family Derivatives

## 8XC552/562 overview



## 80C51 Family Derivatives

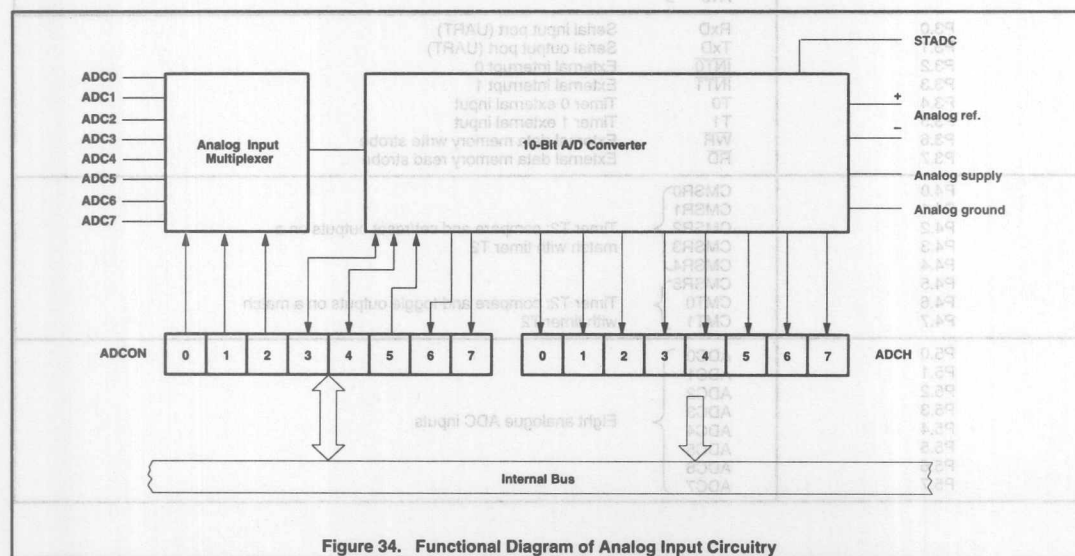
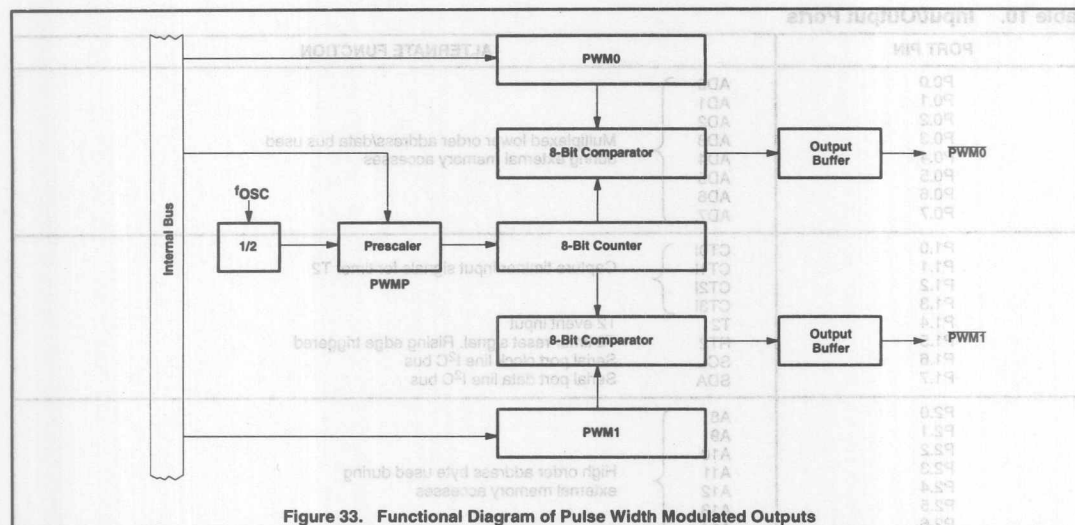
## 8XC552/562 overview

Table 10. Input/Output Ports

PORT PIN	ALTERNATE FUNCTION
P0.0 P0.1 P0.2 P0.3 P0.4 P0.5 P0.6 P0.7	AD0 AD1 AD2 AD3 AD4 AD5 AD6 AD7 <p>Multiplexed lower order address/data bus used during external memory accesses</p>
P1.0 P1.1 P1.2 P1.3 P1.4 P1.5 P1.6 P1.7	CT0I CT1I CT2I CT3I T2 RT2 SCL SDA <p>Capture timer input signals for timer T2</p> <p>T2 event input</p> <p>T2 timer reset signal. Rising edge triggered</p> <p>Serial port clock line I<sup>2</sup>C bus</p> <p>Serial port data line I<sup>2</sup>C bus</p>
P2.0 P2.1 P2.2 P2.3 P2.4 P2.5 P2.6 P2.7	A8 A9 A10 A11 A12 A13 A14 A15 <p>High order address byte used during external memory accesses</p>
P3.0 P3.1 P3.2 P3.3 P3.4 P3.5 P3.6 P3.7	RxD TxD INT0 INT1 T0 T1 WR RD <p>Serial input port (UART)</p> <p>Serial output port (UART)</p> <p>External interrupt 0</p> <p>External interrupt 1</p> <p>Timer 0 external input</p> <p>Timer 1 external input</p> <p>External data memory write strobe</p> <p>External data memory read strobe</p>
P4.0 P4.1 P4.2 P4.3 P4.4 P4.5 P4.6 P4.7	CMSR0 CMSR1 CMSR2 CMSR3 CMSR4 CMSR5 CMT0 CMT1 <p>Timer T2: compare and set/reset outputs on a match with timer T2</p> <p>Timer T2: compare and toggle outputs on a match with timer T2</p>
P5.0 P5.1 P5.2 P5.3 P5.4 P5.5 P5.6 P5.7	ADC0 ADC1 ADC2 ADC3 ADC4 ADC5 ADC6 ADC7 <p>Eight analogue ADC inputs</p>

## 80C51 Family Derivatives

## 8XC552/562 overview





## 80C51 Family Derivatives

## 29V 8XC552/562 overview

**Analog-to-Digital Conversion:** Figure 35 shows the elements of a successive approximation (SA) ADC. The ADC contains a DAC which converts the contents of a successive approximation register to a voltage (VDAC) which is compared to the analog input voltage ( $V_{in}$ ). The output of the comparator is fed to the successive approximation control logic which controls the successive approximation register. A conversion is initiated by setting ADCS in the ADCON register. ADCS can be set by software only or by either hardware or software.

The software only start mode is selected when control bit ADCON.5 (ADEX) = 0. A conversion is then started by setting control bit ADCON.3 (ADCS). The hardware or software start mode is selected when ADCON.5 = 1, and a conversion may be started by setting ADCON.3 as above or by applying a rising edge to external pin STADC. When a conversion is started by applying a rising edge, a low level must be applied to STADC for at least one machine cycle followed by a high level for at least one machine cycle.

The low-to-high transition of STADC is recognized at the end of a machine cycle, and the conversion commences at the beginning of the next cycle. When a conversion is initiated by software, the conversion starts at the beginning of the machine cycle which follows the instruction that sets ADCS. ADCS is actually

implemented with two flip-flops: a command flip-flop which is affected by set operations, and a status flag which is accessed during read operations.

The next two machine cycles are used to initiate the converter. At the end of the first cycle, the ADCS status flag is set and a value of "1" will be returned if the ADCS flag is read while the conversion is in progress. Sampling of the analog input commences at the end of the second cycle.

During the next eight machine cycles, the voltage at the previously selected pin of port 5 is sampled, and this input voltage should be stable in order to obtain a useful sample. In any event, the input voltage slew rate must be less than 10V/ms in order to prevent an undefined result.

The successive approximation control logic first sets the most significant bit and clears all other bits in the successive approximation register (10 0000 0000B). The output of the DAC (50% full scale) is compared to the input voltage  $V_{in}$ . If the input voltage is greater than VDAC, then the bit remains set; otherwise it is cleared.

The successive approximation control logic now sets the next most significant bit (11 0000 0000B or 01 0000 0000B, depending on the previous result), and VDAC is compared to  $V_{in}$  again. If the input voltage is greater than VDAC, then the bit being tested remains set; otherwise the bit being

tested is cleared. This process is repeated until all ten bits have been tested, at which stage the result of the conversion is held in the successive approximation register. Figure 36 shows a conversion flow chart. The bit pointer identifies the bit under test. The conversion takes four machine cycles per bit.

The end of the 10-bit conversion is flagged by control bit ADCON.4 (ADCI). The upper 8 bits of the result are held in special function register ADCH, and the two remaining bits are held in ADCON.7 (ADC.1) and ADCON.6 (ADC.0). The user may ignore the two least significant bits in ADCON and use the ADC as an 8-bit converter (8 upper bits in ADCH). In any event, the total actual conversion time is 50 machine cycles for the 8XC552 or 24 machine cycles for the 8XC562. ADCI will be set and the ADCS status flag will be reset 50 (or 24) cycles after the command flip-flop (ADCS) is set.

Control bits ADCON.0, ADCON.1, and ADCON.2 are used to control an analog multiplexer which selects one of eight analog channels (see Figure 37). An ADCajmp RET mr conversion in progress is unaffected by an external or software ADC start. The result of a completed conversion remains unaffected provided ADCI = logic 1; a new ADC conversion already in progress is aborted when the idle or power-down mode is entered. The result of a completed conversion (ADCI = logic 1) remains unaffected when entering the idle mode.

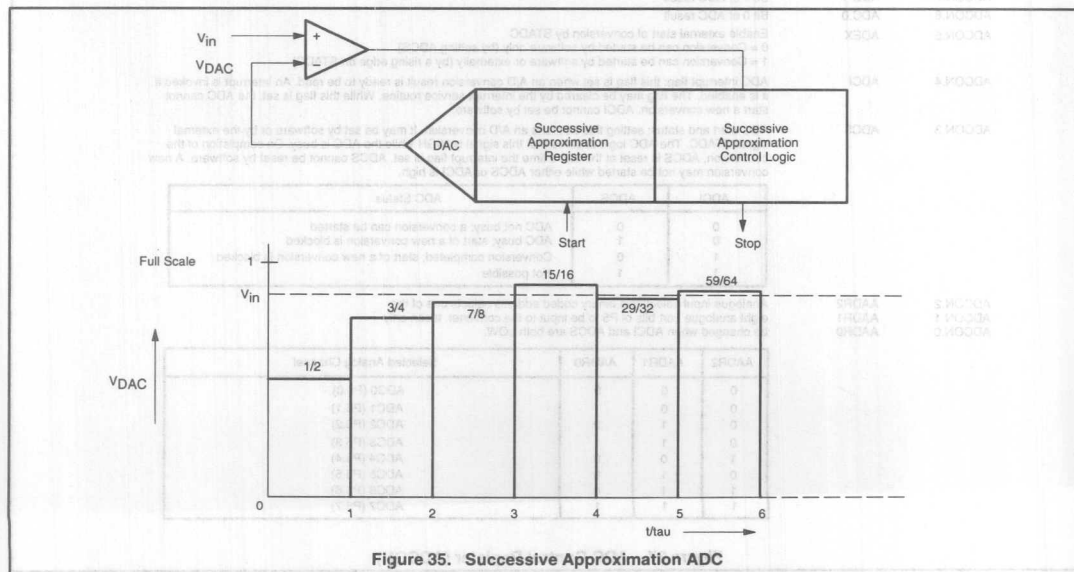


Figure 35. Successive Approximation ADC

80C51 Family Derivatives

8XC552/562 overview

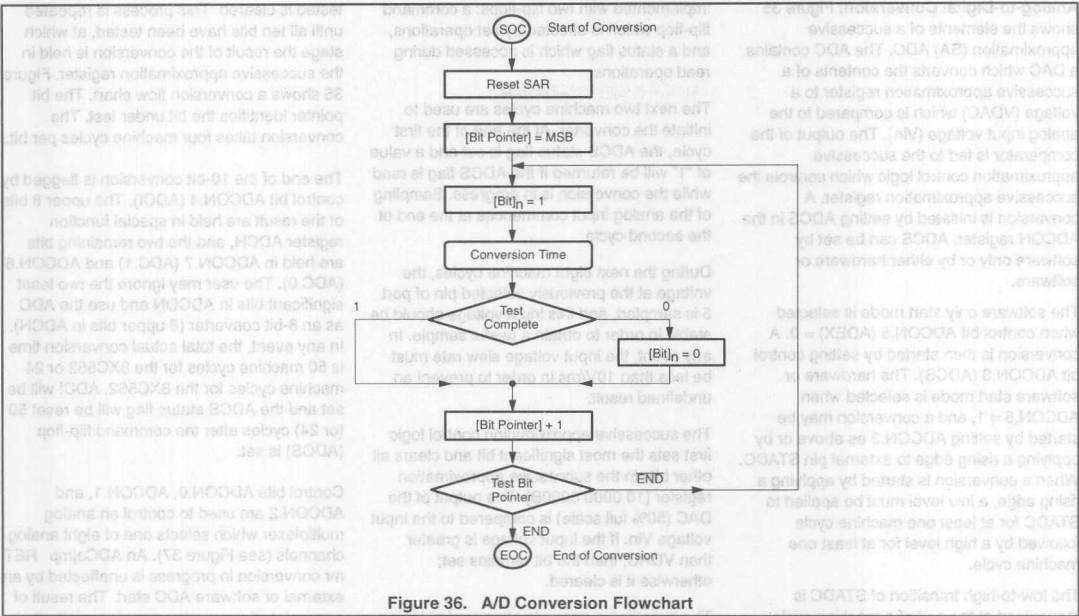


Figure 36. A/D Conversion Flowchart

		7	6	5	4	3	2	1	0																																				
ADCON (CSH)		ADC.1	ADC.0	ADEX	ADCI	ADCS	AADR2	AADR1	AADR0																																				
		(MSB)			(LSB)																																								
Bit	Symbol	Function																																											
ADCON.7	ADC.1	Bit 1 of ADC result																																											
ADCON.6	ADC.0	Bit 0 of ADC result																																											
ADCON.5	ADEX	Enable external start of conversion by STADC 0 = Conversion can be started by software only (by setting ADCS) 1 = Conversion can be started by software or externally (by a rising edge on STADC)																																											
ADCON.4	ADCI	ADC interrupt flag: this flag is set when an A/D conversion result is ready to be read. An interrupt is invoked if it is enabled. The flag may be cleared by the interrupt service routine. While this flag is set, the ADC cannot start a new conversion. ADCI cannot be set by software.																																											
ADCON.3	ADCS	ADC start and status: setting this bit starts an A/D conversion. It may be set by software or by the external signal STADC. The ADC logic ensures that this signal is HIGH while the ADC is busy. On completion of the conversion, ADCS is reset at the same time the interrupt flag is set. ADCS cannot be reset by software. A new conversion may not be started while either ADCS or ADCI is high.																																											
		<table><tr><th>ADCI</th><th>ADCS</th><th>ADC Status</th></tr><tr><td>0</td><td>0</td><td>ADC not busy; a conversion can be started</td></tr><tr><td>0</td><td>1</td><td>ADC busy; start of a new conversion is blocked</td></tr><tr><td>1</td><td>0</td><td>Conversion completed; start of a new conversion is blocked</td></tr><tr><td>1</td><td>1</td><td>Not possible</td></tr></table>								ADCI	ADCS	ADC Status	0	0	ADC not busy; a conversion can be started	0	1	ADC busy; start of a new conversion is blocked	1	0	Conversion completed; start of a new conversion is blocked	1	1	Not possible																					
ADCI	ADCS	ADC Status																																											
0	0	ADC not busy; a conversion can be started																																											
0	1	ADC busy; start of a new conversion is blocked																																											
1	0	Conversion completed; start of a new conversion is blocked																																											
1	1	Not possible																																											
ADCON.2	AADR2	Analogue input select: this binary coded address selects one of the eight analogue port bits of P5 to be input to the converter. It can only be changed when ADCI and ADCS are both LOW.																																											
ADCON.1	AADR1																																												
ADCON.0	AADR0																																												
		<table><tr><th>AADR2</th><th>AADR1</th><th>AADR0</th><th>Selected Analog Channel</th></tr><tr><td>0</td><td>0</td><td>0</td><td>ADC0 (P5.0)</td></tr><tr><td>0</td><td>0</td><td>1</td><td>ADC1 (P5.1)</td></tr><tr><td>0</td><td>1</td><td>0</td><td>ADC2 (P5.2)</td></tr><tr><td>0</td><td>1</td><td>1</td><td>ADC3 (P5.3)</td></tr><tr><td>1</td><td>0</td><td>0</td><td>ADC4 (P5.4)</td></tr><tr><td>0</td><td>1</td><td>0</td><td>ADC5 (P5.5)</td></tr><tr><td>1</td><td>1</td><td>0</td><td>ADC6 (P5.6)</td></tr><tr><td>1</td><td>1</td><td>1</td><td>ADC7 (P5.7)</td></tr></table>								AADR2	AADR1	AADR0	Selected Analog Channel	0	0	0	ADC0 (P5.0)	0	0	1	ADC1 (P5.1)	0	1	0	ADC2 (P5.2)	0	1	1	ADC3 (P5.3)	1	0	0	ADC4 (P5.4)	0	1	0	ADC5 (P5.5)	1	1	0	ADC6 (P5.6)	1	1	1	ADC7 (P5.7)
AADR2	AADR1	AADR0	Selected Analog Channel																																										
0	0	0	ADC0 (P5.0)																																										
0	0	1	ADC1 (P5.1)																																										
0	1	0	ADC2 (P5.2)																																										
0	1	1	ADC3 (P5.3)																																										
1	0	0	ADC4 (P5.4)																																										
0	1	0	ADC5 (P5.5)																																										
1	1	0	ADC6 (P5.6)																																										
1	1	1	ADC7 (P5.7)																																										

Figure 37. ADC Control Register (ADCON)

Figure 37. ADC Control Register (ADCON)

## 80C51 Family Derivatives

## 8XC552/562 overview

**ADC Resolution and Analog Supply:**

Figure 38 shows how the ADC is realized. The ADC has its own supply pins ( $AV_{DD}$  and  $AV_{SS}$ ) and two pins ( $V_{ref+}$  and  $V_{ref-}$ ) connected to each end of the DAC's resistance-ladder. The ladder has 1023 equally spaced taps, separated by a resistance of "R". The first tap is located  $0.5 \times R$  above  $V_{ref-}$ , and the last tap is located  $1.5 \times R$  below  $V_{ref+}$ . This gives a total ladder resistance of  $1024 \times R$ . This structure ensures that the DAC is monotonic and results in a symmetrical quantization error as shown in Figure 40.

For input voltages between  $V_{ref-}$  and  $(V_{ref-}) + 1/2$  LSB, the 10-bit result of an A/D conversion will be 00 0000 0000B = 000H. For input voltages between  $(V_{ref+}) - 3/2$  LSB and  $V_{ref+}$ , the result of a conversion will be 11 1111 1111B = 3FFH.  $AV_{ref+}$  and  $AV_{ref-}$

may be between  $AV_{DD} + 0.2V$  and  $AV_{SS} - 0.2V$ .  $AV_{ref+}$  should be positive with respect to  $AV_{ref-}$ , and the input voltage ( $V_{in}$ ) should be between  $AV_{ref+}$  and  $AV_{ref-}$ . If the analog input voltage range is from 2V to 4V, then 10-bit resolution can be obtained over this range if  $AV_{ref+} = 4V$  and  $AV_{ref-} = 2V$ .

The result can always be calculated from the following formula:

$$\text{Result} = 1024 \times \frac{V_{in} - AV_{ref-}}{AV_{ref+} - AV_{ref-}}$$

**Power Reduction Modes**

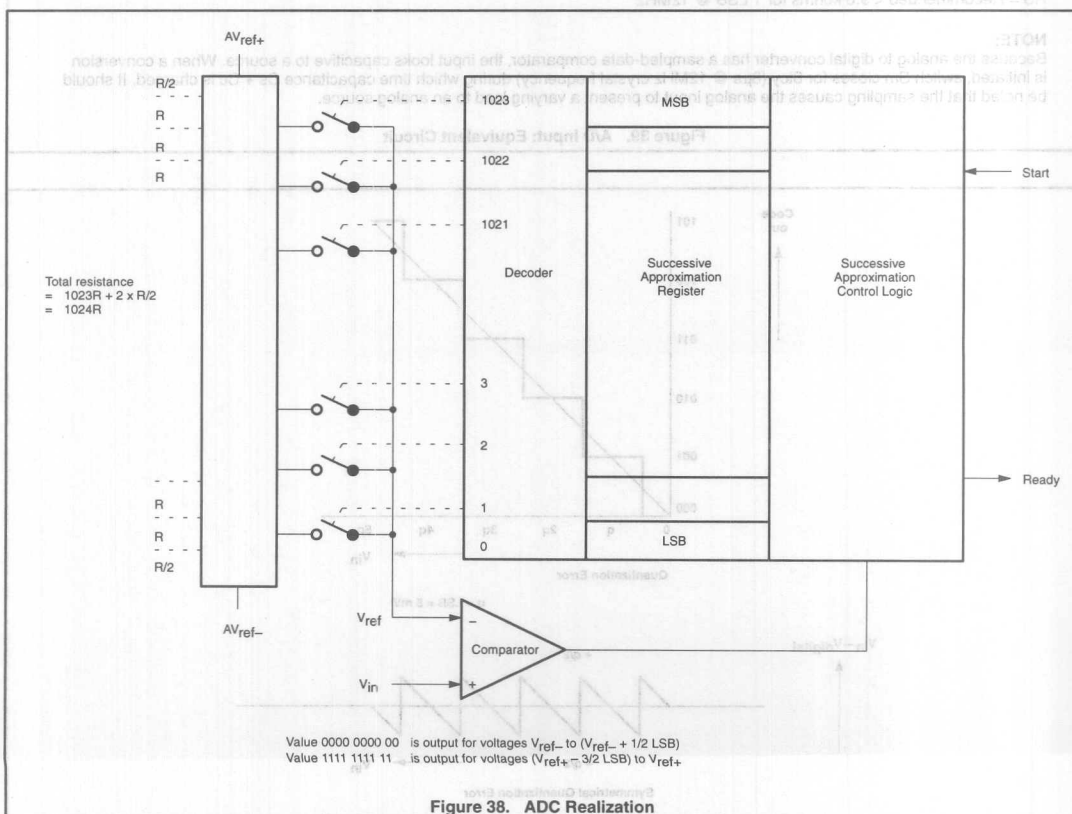
The 8XC552 has two reduced power modes of operation: the idle mode and the power-down mode. These modes are entered by setting bits in the PCON special function register. When the 8XC552 enters the idle mode, the following functions are disabled:

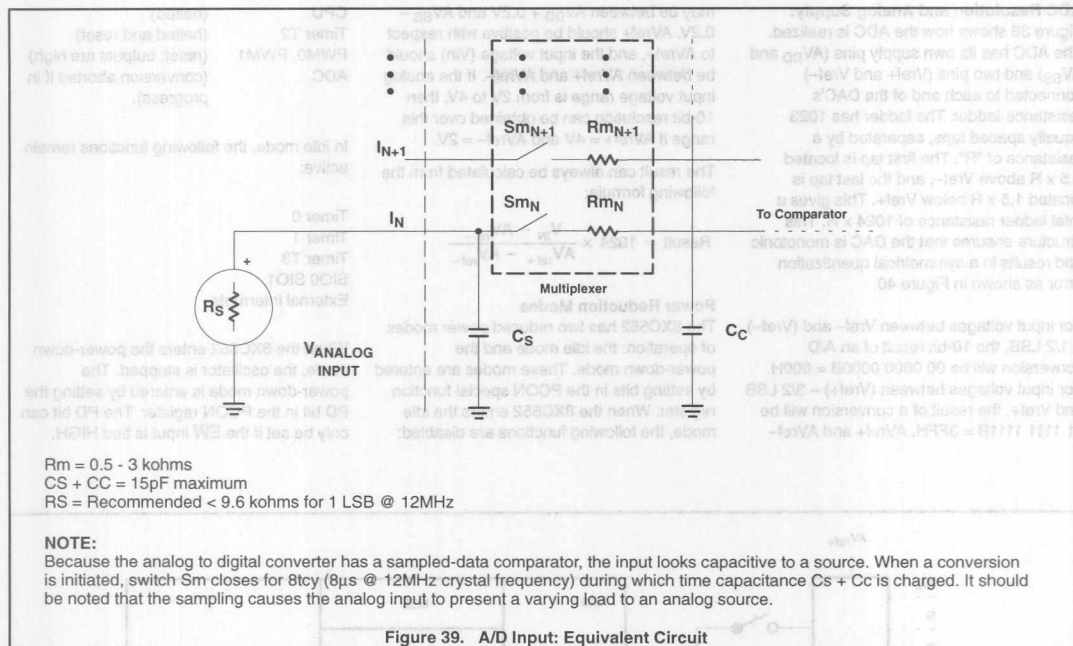
CPU	(halted)
Timer T2	(halted and reset)
PWM0, PWM1	(reset; outputs are high)
ADC	(conversion aborted if in progress).

In idle mode, the following functions remain active:

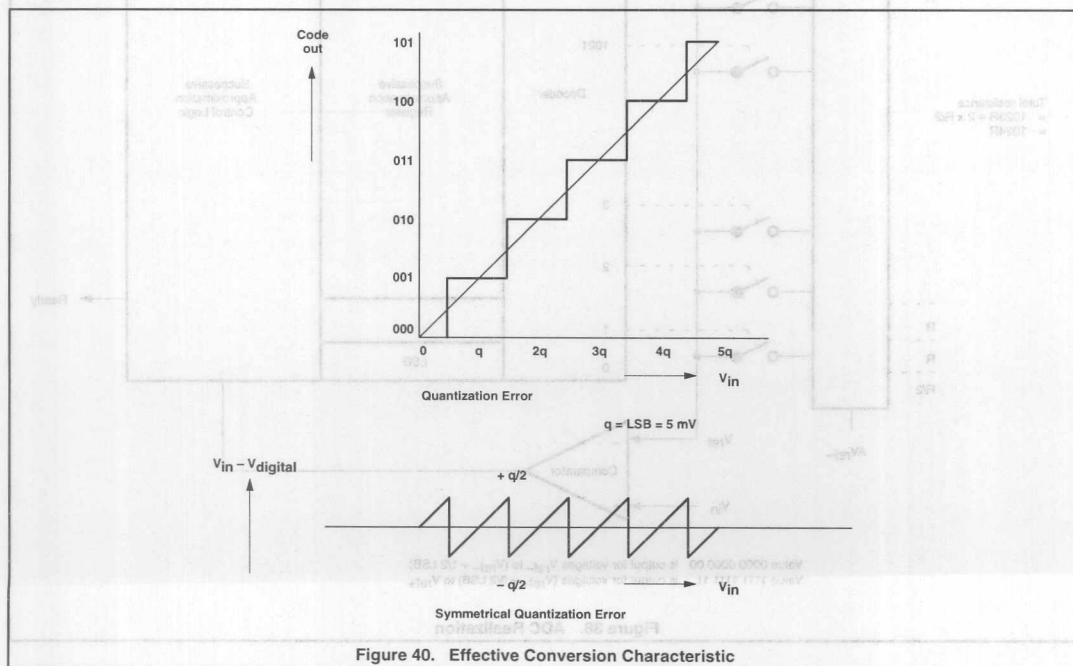
Timer 0
Timer 1
Timer T3
SIO0 SIO1
External interrupts

When the 8XC552 enters the power-down mode, the oscillator is stopped. The power-down mode is entered by setting the PD bit in the PCON register. The PD bit can only be set if the EW input is tied HIGH.





**Figure 39. A/D Input: Equivalent Circuit**



**Figure 40. Effective Conversion Characteristic**

## 80C51 Family Derivatives

## 8XC552/562 overview

**Power-Down Mode:** The instruction that sets PCON.1 will be the last instruction executed in the normal operating mode before the power-down mode is entered. In the power-down mode, the on-chip oscillator is stopped. This freezes all functions; only the on-chip RAM and special function registers are held. The port pins output the contents of their respective special function registers. A hardware reset is the only way to terminate the power-down mode. Reset re-defines all the special function registers, but does not change the on-chip RAM.

In the power-down mode,  $V_{DD}$  and  $AV_{DD}$  can be reduced to minimize power consumption.  $V_{DD}$  and  $AV_{DD}$  must not be reduced before the power-down mode is entered and must be restored to the normal operating voltage before the power-down mode is terminated. The reset that terminates the power-down mode also freezes the oscillator. The reset should not be activated before  $V_{DD}$  and  $AV_{DD}$  are restored to their normal operating level, and must be held active long enough to allow the oscillator to restart and stabilize (normally less than 10ms).

The status of the external pins during power-down is shown in Table 11. If the power-down mode is entered while the 8XC552 is executing out of external program memory, the port data that is held in the P2 special function register is restored to port 2. If a port latch contains a "1", the port pin is held HIGH during the power-down mode by the strong pull-up transistor.

**Power Control Register PCON:** The idle and power-down modes are entered by writing to bits in PCON. PCON is not bit addressable. See Figure 41.

#### Memory Organization

The memory organization of the 8XC552 is the same as in the 80C51, with the exception that the 8XC552 has 8k ROM, 256 bytes RAM, and additional SFRs. Addressing modes are the same in the 8XC552 and the 80C51. Details of the differences are given in the following paragraphs.

In the 8XC552, the lower 8k of the 64k program memory address space is filled by internal ROM. By tying the  $\overline{EA}$  pin high, the processor fetches instructions from internal program ROM. Bus expansion for accessing program memory from 8k upwards is automatic since external instruction fetches occur automatically when the program counter exceeds 8191. If the  $\overline{EA}$  pin is tied low, all program memory fetches are from external memory. The execution speed of the 8XC552 is the same regardless of whether fetches are from external or internal program memory. If all storage is on-chip, then byte location 8191 should be left vacant to prevent an undesired pre-fetch from external program memory address 8192.

Certain locations in program memory are reserved for specific programs. Locations 0000H to 0002H are reserved for the initialization program. Following reset, the CPU always begins execution at locations

0000H. Locations 0003H to 0075H are reserved for the fifteen interrupt request service routines.

Functionally, the internal data memory is the most flexible of the address spaces. The internal data memory space is subdivided into a 256-byte internal data RAM address space and a 128-byte special function register (SFR) address space, as shown in Figure 42.

The internal data RAM address space is 0 to 255. Four 8-bit register banks occupy locations 0 to 31. 128 bit locations of the internal data RAM are accessible through direct addressing. These bits reside in 16 bytes of internal data RAM at locations 20H to 2FH. The stack can be located anywhere in the internal data RAM address space by loading the 8-bit stack pointer. The stack depth may be 256 bytes maximum.

The SFR address space is 128 to 255. All registers except the program counter and the four 8-bit register banks reside in this address space. Memory mapping the SFRs allows them to be accessed as easily as internal RAM, and as such, they can be operated on by most instructions. The 56 SFRs are listed in Figure 43, and their mapping in the SFR address space is shown in Figures 44 and 45. RAM bit addresses are the same as in the 80C51 and are summarized in Figure 46. The special function bit addresses are summarized in Figure 47.

Table 11. External Pin Status During Idle and Power-Down Modes

MODE	MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3	PORT 4	PWM0/PWM1
Idle (1)	Internal	1	1	Port data	Port data	Port data	Port data	Port data	HIGH
Idle (1)	External	1	1	Floating	Port data	Address	Port data	Port data	HIGH
Power-down	Internal	0	0	Port data	Port data	Port data	Port data	Port data	HIGH
Power-down	External	0	0	Floating	Port data	Port data	Port data	Port data	HIGH



## 80C51 Family Derivatives

## 8XC552/562 overview

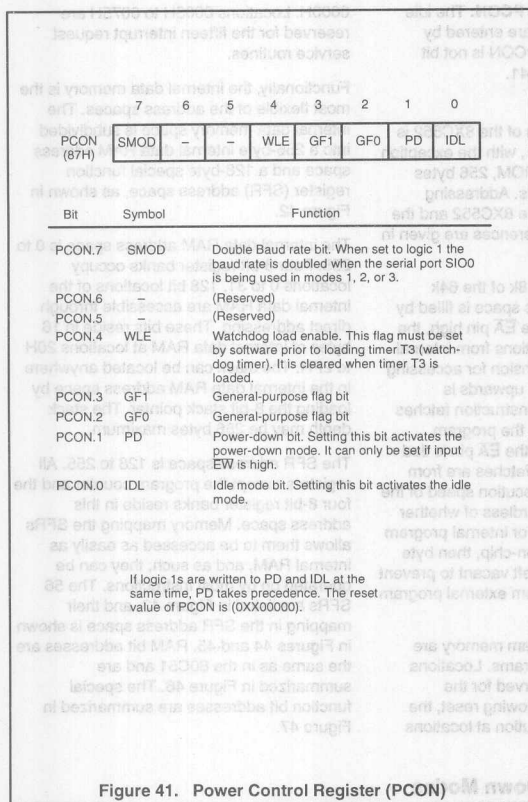


Figure 41. Power Control Register (PCON)

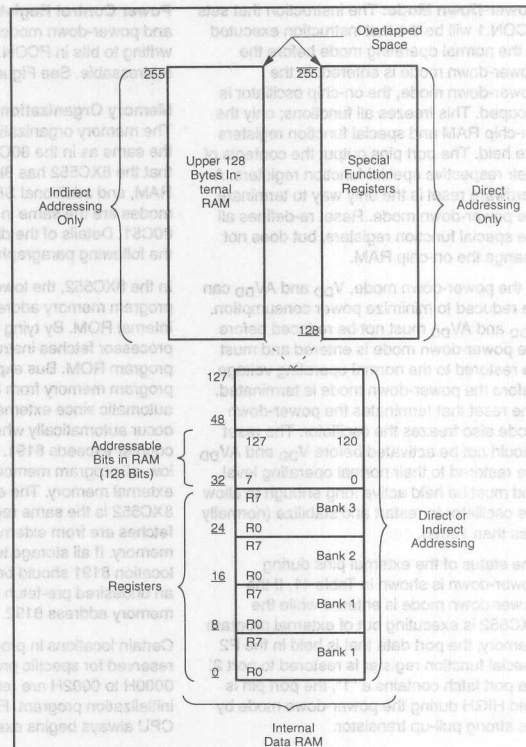


Figure 42. Internal Data Memory Address Space

<b>ARITHMETIC REGISTERS:</b> ACCumulator,* B register,* Program Status Word*	<b>PULSE WIDTH MODULATED O/Ps:</b> Pulse Width Modulation Prescaler Pulse Width Modulation Register 0, Pulse Width Modulation Register 1	<b>CAPTURE AND COMPARE LOGIC:</b> CapTure cONtrol, TiMer T2 Interrupt flag Register, CapTure Low 0, CapTure High 0, CapTure Low 1, CapTure High 1, CapTure Low 2, CapTure High 2, CapTure Low 3, CapTure High 3, CoMpare Low 0, CoMpare High 0, CoMpare Low 1, CoMpare High 1, CoMpare Low 2, CoMpare High 2 SeT Enable, ReseT Enable
<b>POINTERS:</b> Stack Pointer, Data Pointer (High and Low)	<b>SERIAL I/O PORTS:</b> Serial 0 cONtrol,* Serial 0 data BUffler, Serial 1 cONtrol,* Serial 1 DATA, Serial 1 STATus, Serial 1 ADDress, PCON	
<b>PARALLEL I/O PORTS:</b> Port 5,* Port 4,*Port 3,* Port 2,* Port 1,* Port 0*	<b>TIMERS:</b> TiMer MoDE, TiMer cONtrol,* TiMer Low 0, TiMer High 0, TiMer Low 1, TiMer High 1, TiMer T2 cONtrol, TiMer Low 2, TiMer High 2, TiMer T3	<b>ADC</b> ADC cONtrol, ADC High byte

\*NOTE: Bit and byte addressable

Figure 43. Special Function Registers

## 80C51 Family Derivatives

## 8XC552/562 overview

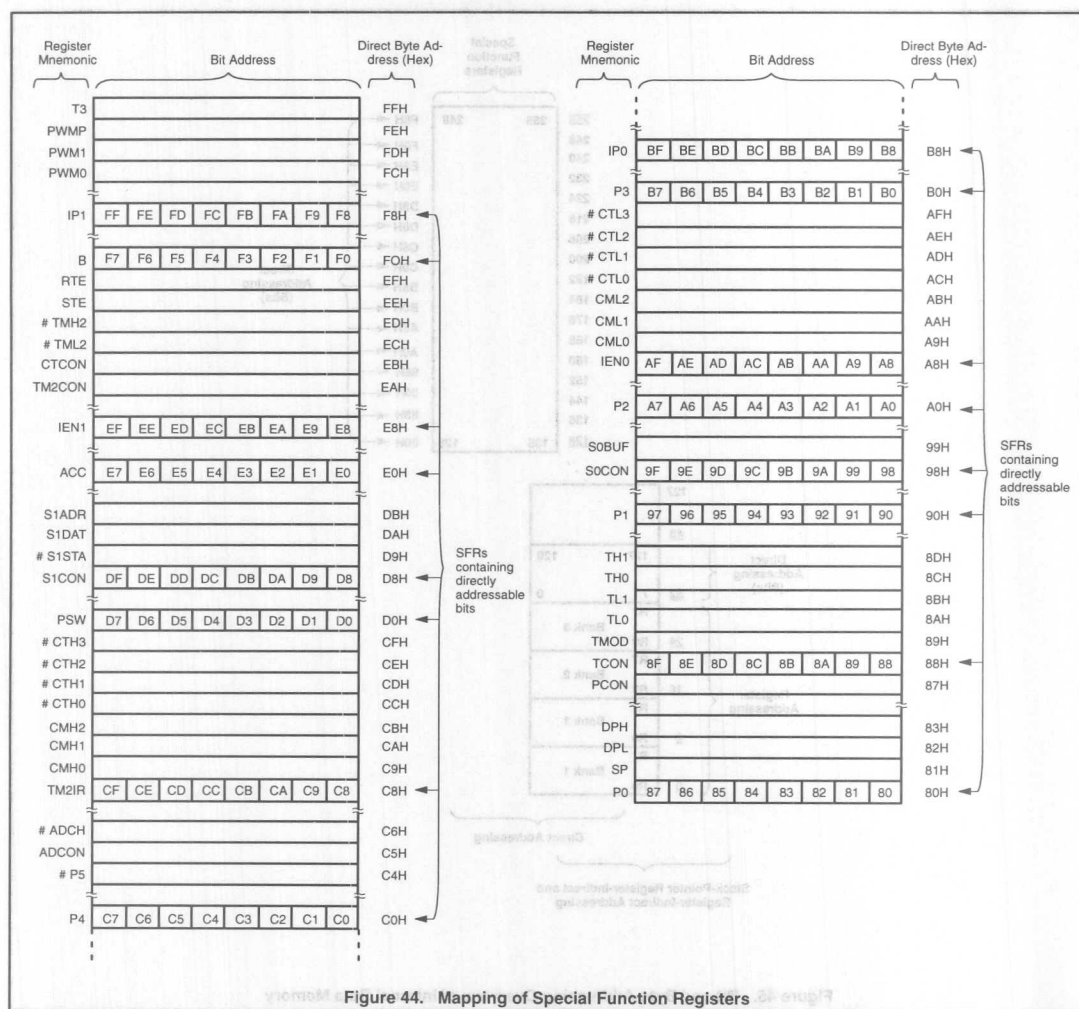


Figure 44. Mapping of Special Function Registers

## 80C51 Family Derivatives

## 8XC552/562 overview

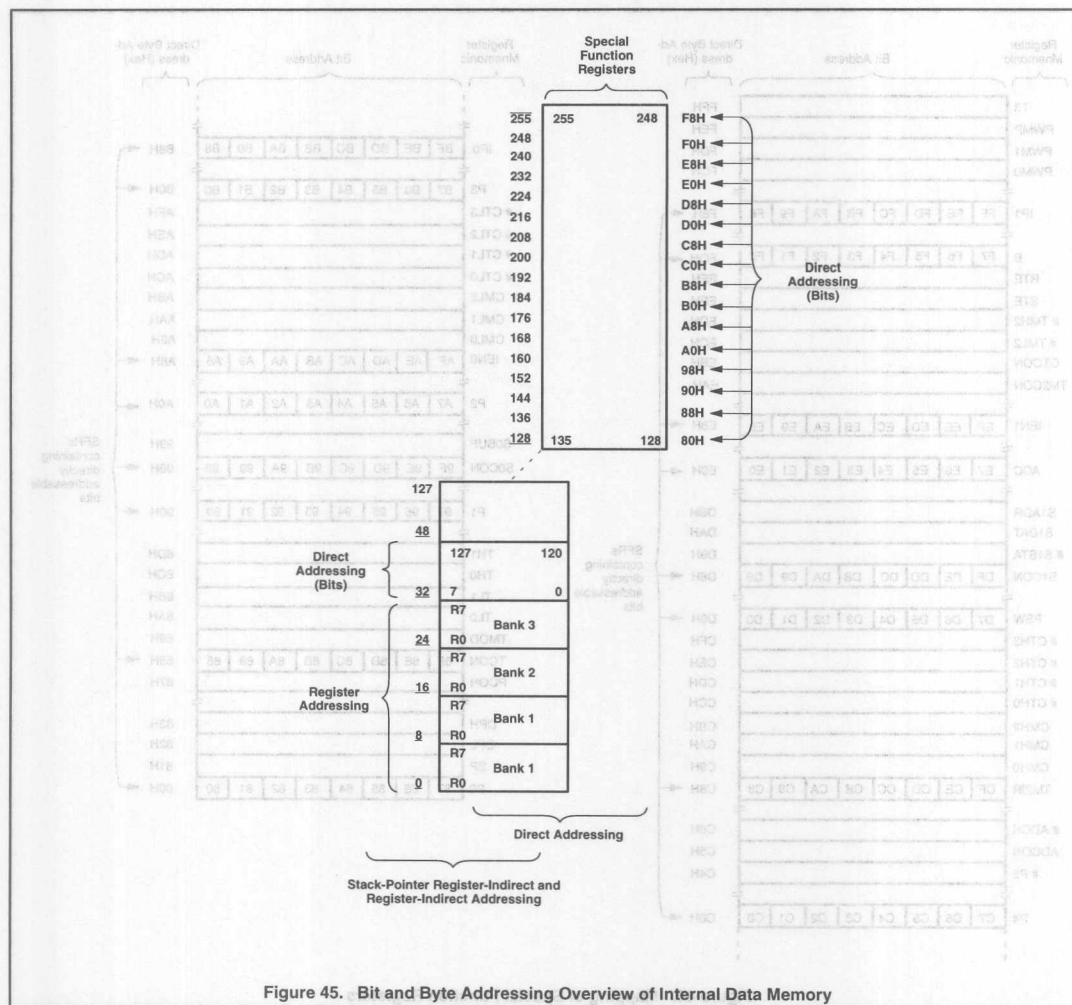


Figure 45. Bit and Byte Addressing Overview of Internal Data Memory

## 80C51 Family Derivatives

## 8XC552/562 overview

7FH	(MSB)	(LSB)	127
2FH	7F	7E	47
2EH	77	76	46
2DH	6F	6E	45
2CH	67	66	44
2BH	5F	5E	43
2AH	57	56	42
29H	4F	4E	41
28H	47	46	40
27H	3F	3E	39
26H	37	36	38
25H	2F	2E	37
24H	27	26	36
23H	1F	1E	35
22H	17	16	34
21H	0F	0E	33
20H	07	06	32
1FH	07	06	31
Bank 3			24
18H			23
17H			16
10H			15
0FH			8
08H			7
07H			0
00H			0

Figure 46. RAM Bit Addresses

Direct Byte Address (Hex)	Bit Address	Register Mnemonic
F8H	PT2 PCM2 PCM1 PCM0 PCT3 PCT2 PCT1 PCT0	IP1
F0H	FF FE FD FC FB FA F9 F8	B
E8H	F7 F6 F5 F4 F3 F2 F1 F0	IEN1
E0H	ET2 ECM2 ECM1 ECM0 ECT3 ECT2 ECT1 ECT0	ACC
D8H	EF EE ED EC EB EA E9 E8	SI CON
D0H	E7 E6 E5 E4 E3 E2 E1 E0	PSW
C8H	CR2 ENS1 STA STO SI AA CR1 CR0	TM2IR
C0H	DF DE DD DC DB DA D9 D8	P4
B8H	CY AC F0 RS1 RS0 OV F1 P	IP0
B0H	D7 D6 D5 D4 D3 D2 D1 D0	P3
A8H	T2OV CM12 CM11 CM10 CT13 CT12 CT11 CT10	IEN0
A0H	CF CE CD CC CB CA C9 C8	P2
98H	C7 C6 C5 C4 C3 C2 C1 C0	SO CON
90H	- PAD PS1 PS0 PT1 PX1 PT0 PX0	P1
88H	BF BE BD BC BB BA B9 B8	TCON
80H	SM0 SM1 SM2 REN TB8 RB8 TI RI	P0
	9F 9E 9D 9C 9B 9A 99 98	
	97 96 95 94 93 92 91 90	
	TF1 TR1 TF0 TR0 IE1 IT1 IE0 IT0	
	8F 8E 8D 8C 8B 8A 89 88	
	87 86 85 84 83 82 81 80	

Figure 47. Special Function Register Bit Address

## Single-chip 8-bit microcontroller

## 80C552/83C552

Single-chip 8-bit microcontroller with 10-bit A/D, capture/compare timer, high-speed outputs, PWM

## DESCRIPTION

The 80C552/83C552 (hereafter generically referred to as 8XC552) Single-Chip 8-Bit Microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 8XC552 has the same instruction set as the 80C51. Three versions of the derivative exist:

- 83C552—8k bytes mask programmable ROM
- 80C552—ROMless version of the 83C552
- 87C552—8k bytes EPROM (described in a separate chapter)

The 8XC552 contains a non-volatile 8k × 8 read-only program memory (83C552), a volatile 256 × 8 read/write data memory, five 8-bit I/O ports, one 8-bit input port, two 16-bit timer/event counters (identical to the timers of the 80C51), an additional 16-bit timer coupled to capture and compare latches, a 15-source, two-priority-level, nested interrupt structure, an 8-input ADC, a dual DAC pulse width modulated interface, two serial interfaces (UART and I<sup>2</sup>C-bus), a "watchdog" timer and on-chip oscillator and timing circuits. For systems that require extra capability, the 8XC552 can be expanded using standard TTL compatible memories and logic.

In addition, the 8XC552 has two software selectable modes of power reduction—idle mode and power-down mode. The idle mode freezes the CPU while allowing the RAM, timers, serial ports, and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.

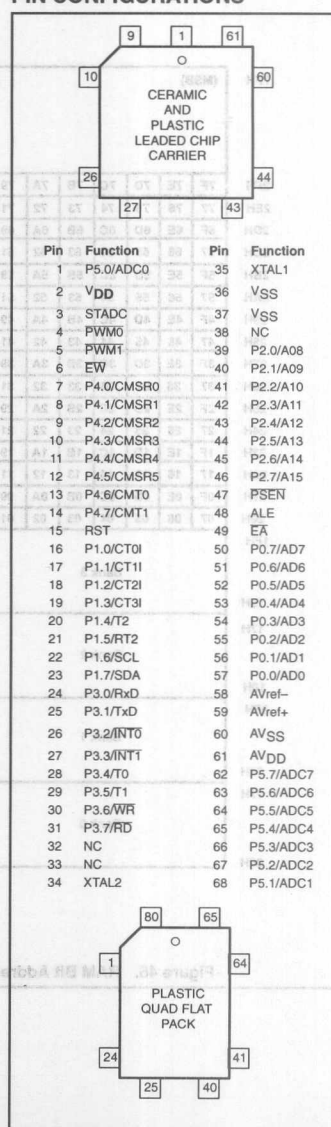
The device also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of over 100 instructions: 49 one-byte, 45 two-byte, and 17 three-byte. With a 16MHz (24MHz) crystal, 58% of the instructions are executed in 0.75µs (0.5µs) and 40% in 1.5µs (1µs). Multiply and divide instructions require 3µs (2µs).



## FEATURES

- 80C51 central processing unit
- 8k × 8 ROM expandable externally to 64k bytes
- An additional 16-bit timer/counter coupled to four capture registers and three compare registers
- Two standard 16-bit timer/counters
- 256 × 8 RAM, expandable externally to 64k bytes
- Capable of producing eight synchronized, timed outputs
- A 10-bit ADC with eight multiplexed analog inputs
- Two 8-bit resolution, pulse width modulation outputs
- Five 8-bit I/O ports plus one 8-bit input port shared with analog inputs
- I<sup>2</sup>C-bus serial I/O port with byte oriented master and slave functions
- Full-duplex UART compatible with the standard 80C51
- On-chip watchdog timer
- Three speed ranges:
  - 1.2 to 16MHz
  - 1.2 to 24MHz (ROM, ROMless only)
  - 1.2 to 30MHz (ROM, ROMless only)
- Three operating ambient temperature ranges:
  - PCB83C552–5: 0°C to +70°C
  - PCF83C552–5: –40°C to +85°C (XTAL frequency max. 24 MHz)
  - PCA83C552–5: –40°C to +125°C (XTAL frequency max. 16 MHz)

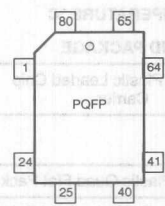
## PIN CONFIGURATIONS





## Single-chip 8-bit microcontroller

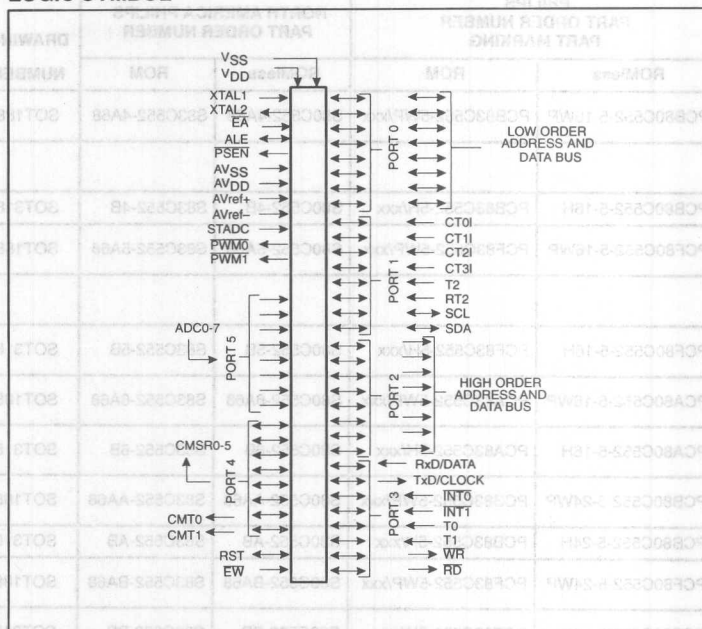
80C552/83C552

PLASTIC QUAD FLAT PACK  
PIN FUNCTIONS


Pin	Function	Pin	Function
1	P4.1/CMSR1	41	P2.3/A11
2	P4.2/CMSR2	42	P2.4/A12
3	NC	43	NC
4	P4.3/CMSR3	44	NC
5	P4.4/CMSR4	45	P2.5/A13
6	P4.5/CMSR5	46	P2.6/A14
7	P4.6/CMT0	47	P2.7/A15
8	P4.7/CMT1	48	PSEN
9	RST	49	ALE
10	P1.0/CT0I	50	EA
11	P1.1/CT1I	51	P0.7/AD7
12	P1.2/CT2I	52	P0.6/AD6
13	P1.3/CT3I	53	P0.5/AD5
14	P1.4/T2	54	P0.4/AD4
15	P1.5/RT2	55	P0.3/AD3
16	P1.6/SCL	56	P0.2/AD2
17	P1.7/SDA	57	P0.1/AD1
18	P3.0/RxD	58	P0.0/AD0
19	P3.1/TxD	59	AVref-
20	P3.2/INT0	60	AVref+
21	NC	61	AVSS
22	NC	62	NC
23	P3.3/INT1	63	AVDD
24	P3.4/T0	64	P5.7/ADC7
25	P3.5/T1	65	P5.6/ADC6
26	P3.6/WR	66	P5.5/ADC5
27	P3.7/RD	67	P5.4/ADC4
28	NC	68	P5.3/ADC3
29	NC	69	P5.2/ADC2
30	NC	70	P5.1/ADC1
31	XTAL2	71	P5.0/ADC0
32	XTAL1	72	VDD
33	IC	73	IC
34	VSS	74	STADC
35	VSS	75	PWM0
36	VSS	76	PWM1
37	NC	77	EW
38	P2.0/A08	78	NC
39	P2.1/A09	79	NC
40	P2.2/A10	80	P4.0/CMSR0

NC = not connected  
IC = internally connected (do not use)

## LOGIC SYMBOL



Single-chip 8-bit microcontroller

80C552/83C552

ORDERING INFORMATION

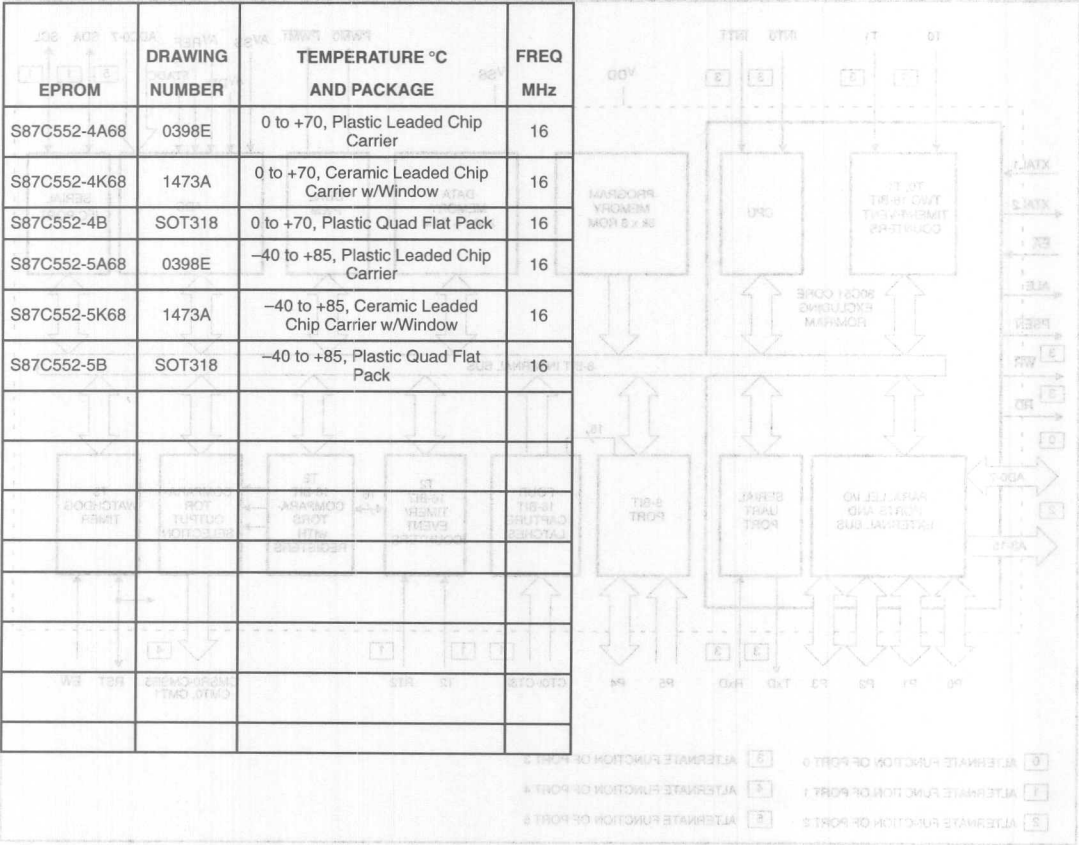
PHILIPS PART ORDER NUMBER PART MARKING		NORTH AMERICA PHILIPS PART ORDER NUMBER		DRAWING NUMBER	TEMPERATURE °C AND PACKAGE	FREQ MHz
ROMless	ROM	ROMless	ROM			
PCB80C552-5-16WP	PCB83C552-5WP/xxx	S80C552-4A68	S83C552-4A68	SOT188	0 to +70, Plastic Leaded Chip Carrier	16
PCB80C552-5-16H	PCB83C552-5H/xxx	S80C552-4B	S83C552-4B	SOT318	0 to +70, Plastic Quad Flat Pack	16
PCF80C552-5-16WP	PCF83C552-5WP/xxx	S80C552-5A68	S83C552-5A68	SOT188	-40 to +85, Plastic Leaded Chip Carrier	16
PCF80C552-5-16H	PCF83C552-5H/xxx	S80C552-5B	S83C552-5B	SOT318	-40 to +85, Plastic Quad Flat Pack	16
PCA80C552-5-16WP	PCA83C552-5WP/xxx	S80C552-6A68	S83C552-6A68	SOT188	-40 to +125, Plastic Leaded Chip Carrier	16
PCA80C552-5-16H	PCA83C552-5H/xxx	S80C552-6B	S83C552-6B	SOT318	-40 to +125, Plastic Quad Flat Pack	16
PCB80C552-5-24WP	PCB83C552-5WP/xxx	S80C552-AA68	S83C552-AA68	SOT188	0 to +70, Plastic Leaded Chip Carrier	24
PCB80C552-5-24H	PCB83C552-5H/xxx	S80C552-AB	S83C552-AB	SOT318	0 to +70, Plastic Quad Flat Pack	24
PCF80C552-5-24WP	PCF83C552-5WP/xxx	S80C552-BA68	S83C552-BA68	SOT188	-40 to +85, Plastic Leaded Chip Carrier	24
PCF80C552-5-24H	PCF83C552-5H/xxx	S80C552-BB	S83C552-BB	SOT318	-40 to +85, Plastic Quad Flat Pack	24
PCB80C552-5-30WP	PCB83C552-5WP/xxx	S80C552-CA68	S83C552-CA68	SOT188	0 to +70, Plastic Leaded Chip Carrier	30
PCB80C552-5-30H	PCB83C552-5H/xxx	S80C552-CB	S83C552-CB	SOT318	0 to +70, Plastic Quad Flat Pack	30

NOTE:

1. xxx denotes the ROM code number.

Single-chip 8-bit microcontroller

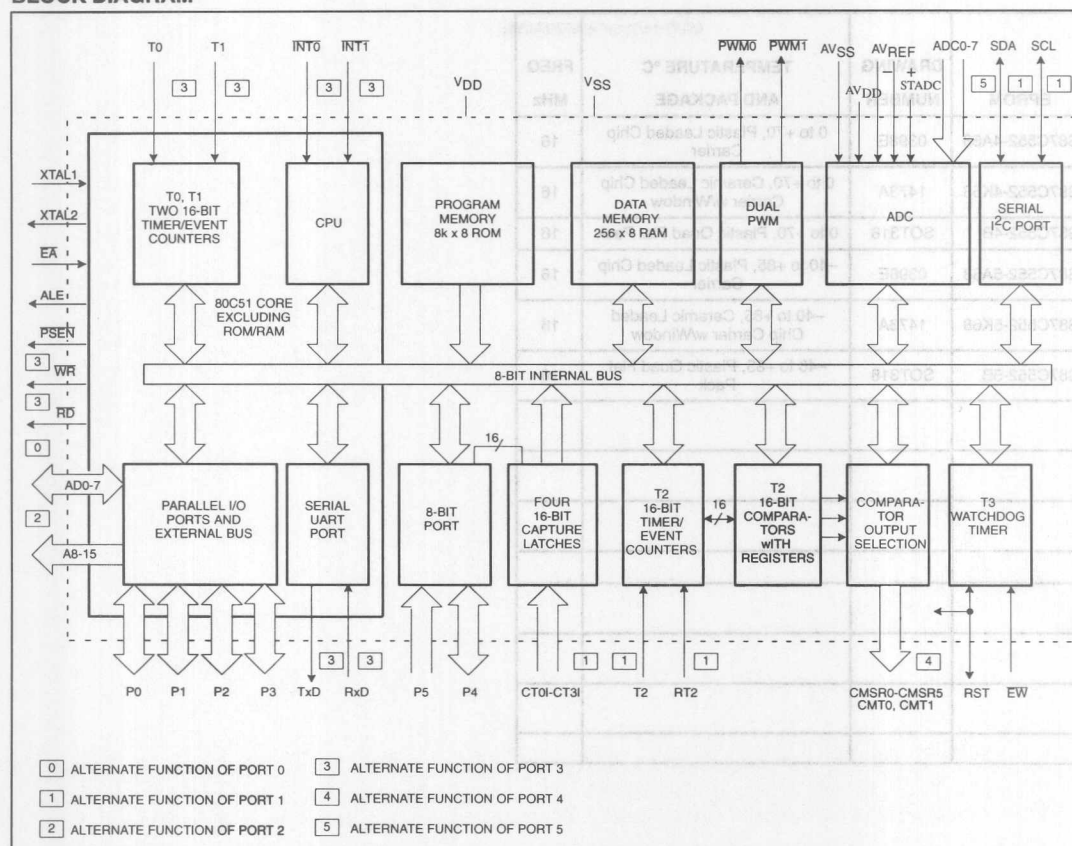
80C552/83C552



## Single-chip 8-bit microcontroller

80C552/83C552

## BLOCK DIAGRAM



## Single-chip 8-bit microcontroller

80C552/83C552

## PIN DESCRIPTION

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	PLCC	QFP		
V <sub>DD</sub>	2	72	I	<b>Digital Power Supply:</b> +5V power supply pin during normal operation, idle and power-down mode.
STADC	3	74	I	<b>Start ADC Operation:</b> Input starting analog to digital conversion (ADC operation can also be started by software). This pin must not float.
PWM0	4	75	O	<b>Pulse Width Modulation:</b> Output 0.
PWM1	5	76	O	<b>Pulse Width Modulation:</b> Output 1.
EW	6	77	I	<b>Enable Watchdog Timer:</b> Enable for T3 watchdog timer and disable power-down mode. This pin must not float.
P0.0-P0.7	57-50	58-51	I/O	<b>Port 0:</b> Port 0 is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application it uses strong internal pull-ups when emitting 1s.
P1.0-P1.7	16-23	10-17	I/O	<b>Port 1:</b> 8-bit I/O port. Alternate functions include: (P1.0-P1.5): Quasi-bidirectional port pins. (P1.6, P1.7): Open drain port pins. <b>CT0I-CT3I (P1.0-P1.3):</b> Capture timer input signals for timer T2. <b>T2 (P1.4):</b> T2 event input. <b>RT2 (P1.5):</b> T2 timer reset signal. Rising edge triggered. <b>SCL (P1.6):</b> Serial port clock line I <sup>2</sup> C-bus. <b>SDA (P1.7):</b> Serial port data line I <sup>2</sup> C-bus. Port 1 is also used to input the lower order address byte during EPROM programming and verification. A0 is on P1.0, etc.
	16-21	10-15	I/O	
	22-23	16-17	I/O	
	16-19	10-13	I	
	20	14	I	
	21	15	I	
	22	16	I/O	
P2.0-P2.7	23	17	I/O	
	39-46	38-42, 45-47	I/O	<b>Port 2:</b> 8-bit quasi-bidirectional I/O port. Alternate function: High-order address byte for external memory (A08-A15).
P3.0-P3.7	24-31	18-20, 23-27	I/O	<b>Port 3:</b> 8-bit quasi-bidirectional I/O port. Alternate functions include: <b>RxD (P3.0):</b> Serial input port. <b>TxD (P3.1):</b> Serial output port. <b>INT0 (P3.2):</b> External interrupt. <b>INT1 (P3.3):</b> External interrupt. <b>T0 (P3.4):</b> Timer 0 external input. <b>T1 (P3.5):</b> Timer 1 external input. <b>WR (P3.6):</b> External data memory write strobe. <b>RD (P3.7):</b> External data memory read strobe.
P4.0-P4.7	7-14	80, 1-2, 4-8	I/O	<b>Port 4:</b> 8-bit quasi-bidirectional I/O port. Alternate functions include: <b>CMSR0-CMSR5 (P4.0-P4.5):</b> Timer T2 compare and set/reset outputs on a match with timer T2. <b>CMT0, CMT1 (P4.6, P4.7):</b> Timer T2 compare and toggle outputs on a match with timer T2.
	7-12	80, 1-2, 4-6	O	
	13, 14	7, 8	O	
P5.0-P5.7	68-62, 1	71-64,	I	<b>Port 5:</b> 8-bit input port. <b>ADC0-ADC7 (P5.0-P5.7):</b> Alternate function: Eight input channels to ADC.
RST	15	9	I/O	<b>Reset:</b> Input to reset the 8XC552. It also provides a reset pulse as output when timer T3 overflows.
XTAL1	35	32	I	<b>Crystal Input 1:</b> Input to the inverting amplifier that forms the oscillator, and input to the internal clock generator. Receives the external clock signal when an external oscillator is used.
XTAL2	34	31	O	<b>Crystal Input 2:</b> Output of the inverting amplifier that forms the oscillator. Left open-circuit when an external clock is used.



## Single-chip 8-bit microcontroller

80C552/83C552

## PIN DESCRIPTION (Continued)

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	PLCC	QFP		
V <sub>SS</sub>	36, 37	34-36	I	Two Digital ground pins.
PSEN	47	48	O	Program Store Enable: Active-low read strobe to external program memory.
ALE	48	49	O	Address Latch Enable: Latches the low byte of the address during accesses to external memory. It is activated every six oscillator periods. During an external data memory access, one ALE pulse is skipped. ALE can drive up to eight LS TTL inputs and handles CMOS inputs without an external pull-up.
EA	49	50	I	External Access: When EA is held at TTL level high, the CPU executes out of the internal program ROM provided the program counter is less than 8192. When EA is held at TTL low level, the CPU executes out of external program memory. EA is not allowed to float.
AV <sub>REF-</sub>	58	59	I	Analog to Digital Conversion Reference Resistor: Low-end.
AV <sub>REF+</sub>	59	60	I	Analog to Digital Conversion Reference Resistor: High-end.
AV <sub>SS</sub>	60	61	I	Analog Ground
AV <sub>DD</sub>	61	63	I	Analog Power Supply

## NOTE:

1. To avoid "latch-up" effect at power-on, the voltage on any pin at any time must not be higher or lower than V<sub>DD</sub> + 0.5V or V<sub>SS</sub> - 0.5V, respectively.

### OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator, as shown in the logic symbol, page 635.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

### RESET

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. To insure a good power-on reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few milliseconds) plus two machine cycles. At power-on, the voltage on V<sub>DD</sub> and RST must come up at the same time for a proper start-up.

### IDLE MODE

In the idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers

remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

### POWER-DOWN MODE

In the power-down mode, the oscillator is stopped and the instruction to invoke power-down is the last instruction executed. Only the contents of the on-chip RAM are preserved. A hardware reset is the only way to terminate the power-down mode. The control bits for the reduced power modes are in the special function register PCON. Table 1 shows the state of the I/O ports during low current operating modes.

Table 1. External Pin Status During Idle and Power-Down Modes

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3	PORT 4	PWM0/PWM1
Idle	Internal	1	1	Data	Data	Data	Data	Data	1
Idle	External	1	1	Float	Data	Address	Data	Data	1
Power-down	Internal	0	0	Data	Data	Data	Data	Data	1
Power-down	External	0	0	Float	Data	Data	Data	Data	1

## Single-chip 8-bit microcontroller

80C552/83C552

## Serial Control Register (S1CON) – See Table 2

S1CON (D8H)	CR2	ENS1	STA	STO	SI	AA	CR1	CR0
-------------	-----	------	-----	-----	----	----	-----	-----

Bits CR0, CR1 and CR2 determine the serial clock frequency that is generated in the master mode of operation.

Table 2. Serial Clock Rates

CR2	CR1	CR0	BIT FREQUENCY (kHz) AT f <sub>osc</sub>					f <sub>osc</sub> DIVIDED BY
			6MHz	12MHz	16MHz	24MHz <sup>2</sup>	30MHz <sup>2</sup>	
0	0	0	23	47	62.5	94	117	256
0	0	1	27	54	71	107	134	224
0	1	0	31	63	83.3	125	156	192
0	1	1	37	75	100	150	188	160
1	0	0	6.25	12.5	17	25	31	960
1	0	1	50	100	133	200	250	120
1	1	0	100	200	267	400	500	60
1	1	1	0.24 < 62.5	0.49 < 62.5	0.65 < 55.6	0.98 < 50.0	1.22 < 52.1	96 × (256 – (reload value Timer 1))
			0 < 255	0 < 254	0 < 253	0 < 251	0 < 250	reload value Timer 1 in Mode 2.

## NOTES:

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
- At f<sub>osc</sub> = 24MHz/ 30MHz the maximum I<sup>2</sup>C bus rate of 100kHz cannot be realized due to the fixed divider rates.

ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>

PARAMETER	RATING	UNIT
Storage temperature range	–65 to +150	°C
Voltage on any other pin to V <sub>SS</sub>	–0.5 to +6.5	V
Input, output DC current on any single I/O pin	5.0	mA
Power dissipation (based on package heat transfer limitations, not device power consumption)	1.0	W

## NOTES:

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
- This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
- Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V<sub>SS</sub> unless otherwise noted.

## DEVICE SPECIFICATIONS

TYPE	SUPPLY VOLTAGE (V)		FREQUENCY (MHz)		TEMPERATURE RANGE (°C)
	MIN	MAX	MIN	MAX	
PCB83(0)C552-5-16	4.0	6.0	1.2	16	0 to +70
PCF83(0)C552-5-16	4.0	6.0	1.2	16	–40 to +85
PCA83(0)C552-5-16	4.5	5.5	1.2	16	–40 to +125
PCB83(0)C552-5-24	4.5	5.5	1.2	24	0 to +70
PCF83(0)C552-5-24	4.5	5.5	1.2	24	–40 to +85
PCB83(0)C552-5-30	4.5	5.5	1.2	30	0 to +70

## Single-chip 8-bit microcontroller

## 80C552/83C552

## DC ELECTRICAL CHARACTERISTICS

 $V_{SS}, AV_{SS} = 0V$ 

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			MIN	MAX	
$I_{DD}$	Supply current operating:	See notes 1 and 2			
	PCB8XC552-5-16	$f_{osc} = 16MHz$		45	mA
	PCF8XC552-5-16	$f_{osc} = 16MHz$		45	mA
	PCA8XC552-5-16	$f_{osc} = 16MHz$		40	mA
	PCB8XC552-5-24	$f_{osc} = 24MHz$		55	mA
	PCF8XC552-5-24	$f_{osc} = 24MHz$		55	mA
	PCB8XC552-5-30	$f_{osc} = 30MHz$		68	mA
$I_{ID}$	Idle mode:	See notes 1 and 3			
	PCB8XC552-5-16	$f_{osc} = 16MHz$		10	mA
	PCF8XC552-5-16	$f_{osc} = 16MHz$		10	mA
	PCA8XC552-5-16	$f_{osc} = 16MHz$		9	mA
	PCB8XC552-5-24	$f_{osc} = 24MHz$		12.5	mA
	PCF8XC552-5-24	$f_{osc} = 24MHz$		12.5	mA
	PCB8XC552-5-30	$f_{osc} = 30MHz$		15	mA
$I_{PD}$	Power-down current:	See notes 1 and 4; $2V < V_{PD} < V_{DDmax}$			
	PCB8XC552			50	$\mu A$
	PCF8XC552			50	$\mu A$
	PCA8XC552			150	$\mu A$
<b>Inputs</b>					
$V_{IL}$	Input low voltage, except $\overline{EA}$ , P1.6, P1.7		-0.5	$0.2V_{DD}-0.1$	V
$V_{IL1}$	Input low voltage to $\overline{EA}$		-0.5	$0.2V_{DD}-0.3$	V
$V_{IL2}$	Input low voltage to P1.6/SCL, P1.7/SDA <sup>5</sup>		-0.5	$0.3V_{DD}$	V
$V_{IH}$	Input high voltage, except XTAL1, RST, P1.6/SCL, P1.7/SDA		$0.2V_{DD}+0.9$	$V_{DD}+0.5$	V
$V_{IH1}$	Input high voltage, XTAL1, RST		$0.7V_{DD}$	$V_{DD}+0.5$	V
$V_{IH2}$	Input high voltage, P1.6/SCL, P1.7/SDA <sup>5</sup>		$0.7V_{DD}$	6.0	V
$I_{IL}$	Logical 0 input current, ports 1, 2, 3, 4, except P1.6, P1.7	$V_{IN} = 0.45V$		-50	$\mu A$
$I_{TL}$	Logical 1-to-0 transition current, ports 1, 2, 3, 4, except P1.6, P1.7	See note 6		-650	$\mu A$
$\pm I_{IL1}$	Input leakage current, port 0, $\overline{EA}$ , STADC, EW	$0.45V < V_I < V_{DD}$		10	$\mu A$
$\pm I_{IL2}$	Input leakage current, P1.6/SCL, P1.7/SDA	$0V < V_I < 6V$ $0V < V_{DD} < 5.5V$		10	$\mu A$
$\pm I_{IL3}$	Input leakage current, port 5	$0.45V < V_I < V_{DD}$		1	$\mu A$
<b>Outputs</b>					
$V_{OL}$	Output low voltage, ports 1, 2, 3, 4, except P1.6, P1.7	$I_{OL} = 1.6mA^7$		0.45	V
$V_{OL1}$	Output low voltage, port 0, ALE, PSEN, PWM0, PWM1	$I_{OL} = 3.2mA^7$		0.45	V
$V_{OL2}$	Output low voltage, P1.6/SCL, P1.7/SDA	$I_{OL} = 3.0mA^7$		0.4	V
$V_{OH}$	Output high voltage, ports 1, 2, 3, 4, except P1.6/SCL, P1.7/SDA	$V_{DD} = 5V + 10\%$			
		$-I_{OH} = 60\mu A$	2.4		V
		$-I_{OH} = 25\mu A$	$0.75V_{DD}$		V
		$-I_{OH} = 10\mu A$	$0.9V_{DD}$		V

## Single-chip 8-bit microcontroller

80C552/83C552

## DC ELECTRICAL CHARACTERISTICS (Continued)

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			MIN	MAX	
Outputs (Continued)					
V <sub>OH1</sub>	Output high voltage (port 0 in external bus mode, ALE, PSEN, PWM0, PWM1) <sup>8</sup>	V <sub>DD</sub> = 5V +10% -I <sub>OH</sub> = 400μA -I <sub>OH</sub> = 150μA -I <sub>OH</sub> = 40μA	2.4 0.75V <sub>DD</sub> 0.9V <sub>DD</sub>		V V V
V <sub>OH2</sub>	Output high voltage (RST)	-I <sub>OH</sub> = 400μA -I <sub>OH</sub> = 120μA	2.4 0.8V <sub>DD</sub>		V V
R <sub>RST</sub>	Internal reset pull-down resistor		50	150	kΩ
C <sub>IO</sub>	Pin capacitance	Test freq = 1MHz, T <sub>amb</sub> = 25°C		10	pF
Analog Inputs					
AV <sub>DD</sub>	Analog supply voltage: PCB8XC552-5-16 PCF8XC552-5-16 PCA8XC552-5-16 PCB8XC552-5-24 PCF8XC552-5-24 PCA8XC552-5-24 PCB8XC552-5-30	AV <sub>DD</sub> = V <sub>DD</sub> ±0.2V AV <sub>DD</sub> = V <sub>DD</sub> ±0.2V AV <sub>DD</sub> = V <sub>DD</sub> ±0.2V AV <sub>DD</sub> = V <sub>DD</sub> ±0.2V AV <sub>DD</sub> = V <sub>DD</sub> ±0.2V AV <sub>DD</sub> = V <sub>DD</sub> ±0.2V AV <sub>DD</sub> = V <sub>DD</sub> ±0.2V	4.0 4.0 4.5 4.5 4.5 4.5 4.5	6.0 6.0 5.5 5.5 5.5 5.5 5.5	V V V V V V V
AI <sub>DD</sub>	Analog supply current: operating: (16MHz) Analog supply current: operating: (24MHz, 30MHz)	Port 5 = 0 to AV <sub>DD</sub> Port 5 = 0 to AV <sub>DD</sub>		1.2 1.0	mA mA
AI <sub>ID</sub>	Idle mode: PCB8XC552-5-16 PCF8XC552-5-16 PCA8XC552-5-16 PCB8XC552-5-24 PCF8XC552-5-24 PCA8XC552-5-24 PCB8XC552-5-30			50 50 100 50 50 50 50	μA μA μA μA μA μA μA
AI <sub>PD</sub>	Power-down mode: PCB8XC552 PCF8XC552 PCA8XC552	2V < AV <sub>PD</sub> < AV <sub>DD</sub> max		50 50 100	μA μA μA
AV <sub>IN</sub>	Analog input voltage		AV <sub>SS</sub> -0.2	AV <sub>DD</sub> +0.2	V
AV <sub>REF</sub>	Reference voltage: AV <sub>REF-</sub> AV <sub>REF+</sub>		AV <sub>SS</sub> -0.2	AV <sub>DD</sub> +0.2	V V
R <sub>REF</sub>	Resistance between AV <sub>REF+</sub> and AV <sub>REF-</sub>		10	50	kΩ
C <sub>IA</sub>	Analog input capacitance			15	pF
t <sub>ADS</sub>	Sampling time			8t <sub>CY</sub>	μs
t <sub>ADC</sub>	Conversion time (including sampling time)			50t <sub>CY</sub>	μs
DL <sub>e</sub>	Differential non-linearity <sup>10, 11, 12</sup>			±1	LSB
IL <sub>e</sub>	Integral non-linearity <sup>10, 13</sup>			±2	LSB
OS <sub>e</sub>	Offset error <sup>10, 14</sup>			±2	LSB

## Single-chip 8-bit microcontroller

80C552/83C552

## DC ELECTRICAL CHARACTERISTICS (Continued)

SYMBOL	PARAMETER	LIMITS		UNIT
		MIN	MAX	
Analog Inputs (continued)				
G <sub>e</sub>	Gain error <sup>10, 15</sup>		±0.4	%
A <sub>e</sub>	Absolute voltage error <sup>10, 16</sup>		±3	LSB
M <sub>CTC</sub>	Channel to channel matching		±1	LSB
C <sub>t</sub>	Crosstalk between inputs of port 5 <sup>17</sup>	0–100kHz	–60	dB

## NOTES FOR DC ELECTRICAL CHARACTERISTICS:

- See Figures 10 through 15 for  $I_{DD}$  test conditions.
- The operating supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 10\text{ns}$ ;  $V_{IL} = V_{SS} + 0.5\text{V}$ ;  $V_{IH} = V_{DD} - 0.5\text{V}$ ; XTAL2 not connected;  $\overline{EA} = \text{RST} = \text{Port } 0 = \text{EW} = V_{DD}$ ; STADC =  $V_{SS}$ .
- The idle mode supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 10\text{ns}$ ;  $V_{IL} = V_{SS} + 0.5\text{V}$ ;  $V_{IH} = V_{DD} - 0.5\text{V}$ ; XTAL2 not connected; Port 0 =  $\text{EW} = V_{DD}$ ;  $\overline{EA} = \text{RST} = \text{STADC} = V_{SS}$ .
- The power-down current is measured with all output pins disconnected; XTAL2 not connected; Port 0 =  $\text{EW} = V_{DD}$ ;  $\overline{EA} = \text{RST} = \text{STADC} = \text{XTAL1} = V_{SS}$ .
- The input threshold voltage of P1.6 and P1.7 (SIO1) meets the I<sup>2</sup>C specification, so an input voltage below 1.5V will be recognized as a logic 0 while an input voltage above 3.0V will be recognized as a logic 1.
- Pins of ports 1 (except P1.6, P1.7), 2, 3, and 4 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when  $V_{IN}$  is approximately 2V.
- Capacitive loading on ports 0 and 2 may cause spurious noise to be superimposed on the  $V_{OL}$ s of ALE and ports 1 and 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100pF), the noise pulse on the ALE pin may exceed 0.8V. In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.  $I_{OL}$  can exceed these conditions provided that no single output sinks more than 5mA and no more than two outputs exceed the test conditions.
- Capacitive loading on ports 0 and 2 may cause the  $V_{OH}$  on ALE and PSEN to momentarily fall below the 0.9 $V_{DD}$  specification when the address bits are stabilizing.
- The following condition must not be exceeded:  $V_{DD} - 0.2\text{V} < AV_{DD} < V_{DD} + 0.2\text{V}$ .
- Conditions:  $AV_{REF-} = 0\text{V}$ ;  $AV_{DD} = 5.0\text{V}$ ,  $AV_{REF+}$  (80C552, 83C552) = 5.12V. ADC is monotonic with no missing codes. Measurement by continuous conversion of  $AV_{IN} = -20\text{mV}$  to 5.12V in steps of 0.5mV.
- The differential non-linearity ( $DL_e$ ) is the difference between the actual step width and the ideal step width. (See Figure 1.)
- The ADC is monotonic; there are no missing codes.
- The integral non-linearity ( $IL_e$ ) is the peak difference between the center of the steps of the actual and the ideal transfer curve after appropriate adjustment of gain and offset error. (See Figure 1.)
- The offset error ( $OS_e$ ) is the absolute difference between the straight line which fits the actual transfer curve (after removing gain error), and a straight line which fits the ideal transfer curve. (See Figure 1.)
- The gain error ( $G_e$ ) is the relative difference in percent between the straight line fitting the actual transfer curve (after removing offset error), and the straight line which fits the ideal transfer curve. Gain error is constant at every point on the transfer curve. (See Figure 1.)
- The absolute voltage error ( $A_e$ ) is the maximum difference between the center of the steps of the actual transfer curve of the non-calibrated ADC and the ideal transfer curve.
- This should be considered when both analog and digital signals are simultaneously input to port 5.

$AV_{REF-}$	Reference voltage			
$AV_{REF+}$	Reference voltage			
$AV_{DD}$	Reference voltage			
$R_{REF}$	Resistance between $AV_{REF-}$ and $AV_{REF+}$	10		k $\Omega$
$C_A$	Analog input capacitance	15		pF
$t_{AS}$	Sampling time	80		$\mu\text{s}$
$t_{AC}$	Conversion time (including sampling time)	500		$\mu\text{s}$
$DL_e$	Differential non-linearity	$\pm 1$		LSB
$IL_e$	Integral non-linearity	$\pm 5$		LSB
$OS_e$	Offset error	$\pm 5$		LSB



## Single-chip 8-bit microcontroller

80C552/83C552

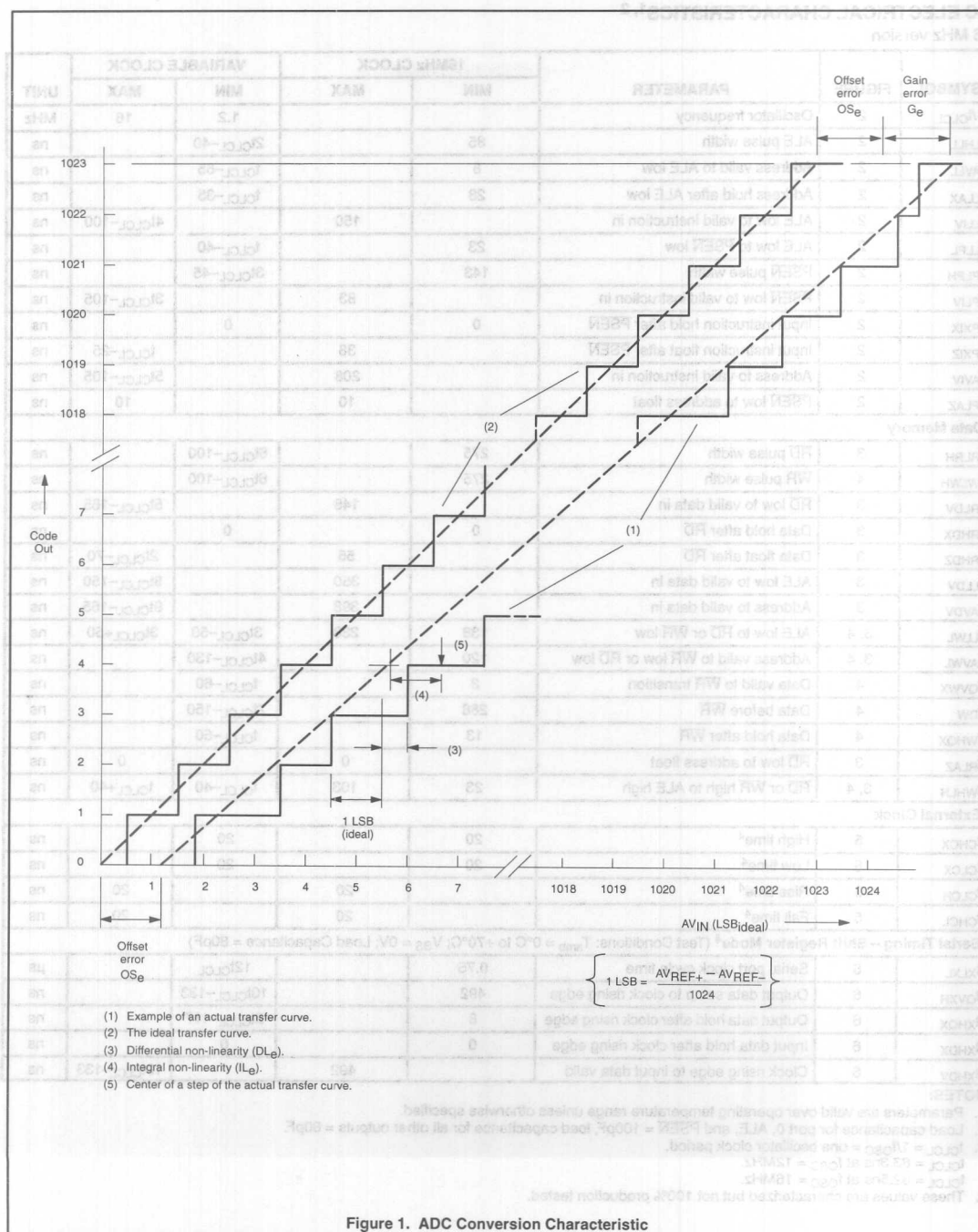


Figure 1. ADC Conversion Characteristic

## Single-chip 8-bit microcontroller

80C552/83C552

AC ELECTRICAL CHARACTERISTICS<sup>1, 2</sup>

16 MHz version

SYMBOL	FIGURE	PARAMETER	16MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
$1/t_{CLCL}$	2	Oscillator frequency			1.2	16	MHz
$t_{HLH}$	2	ALE pulse width	85		$2t_{CLCL}-40$		ns
$t_{AVLL}$	2	Address valid to ALE low	8		$t_{CLCL}-55$		ns
$t_{LLAX}$	2	Address hold after ALE low	28		$t_{CLCL}-35$		ns
$t_{LLIV}$	2	ALE low to valid instruction in		150		$4t_{CLCL}-100$	ns
$t_{LLPL}$	2	ALE low to PSEN low	23		$t_{CLCL}-40$		ns
$t_{PLPH}$	2	PSEN pulse width	143		$3t_{CLCL}-45$		ns
$t_{PLIV}$	2	PSEN low to valid instruction in		83		$3t_{CLCL}-105$	ns
$t_{PXIX}$	2	Input instruction hold after PSEN	0		0		ns
$t_{PXIZ}$	2	Input instruction float after PSEN		38		$t_{CLCL}-25$	ns
$t_{AVIV}$	2	Address to valid instruction in		208		$5t_{CLCL}-105$	ns
$t_{PLAZ}$	2	PSEN low to address float		10		10	ns
<b>Data Memory</b>							
$t_{RLRH}$	3	RD pulse width	275		$6t_{CLCL}-100$		ns
$t_{WLWH}$	4	WR pulse width	275		$6t_{CLCL}-100$		ns
$t_{RLDV}$	3	RD low to valid data in		148		$5t_{CLCL}-165$	ns
$t_{RHDX}$	3	Data hold after RD	0		0		ns
$t_{RHDZ}$	3	Data float after RD		55		$2t_{CLCL}-70$	ns
$t_{LLDV}$	3	ALE low to valid data in		350		$8t_{CLCL}-150$	ns
$t_{AVDV}$	3	Address to valid data in		398		$9t_{CLCL}-165$	ns
$t_{LLWL}$	3, 4	ALE low to RD or WR low	138	238	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
$t_{AVWL}$	3, 4	Address valid to WR low or RD low	120		$4t_{CLCL}-130$		ns
$t_{QVWX}$	4	Data valid to WR transition	3		$t_{CLCL}-60$		ns
$t_{DW}$	4	Data before WR	288		$7t_{CLCL}-150$		ns
$t_{WHQX}$	4	Data hold after WR	13		$t_{CLCL}-50$		ns
$t_{RLAZ}$	3	RD low to address float		0		0	ns
$t_{WHLH}$	3, 4	RD or WR high to ALE high	23	103	$t_{CLCL}-40$	$t_{CLCL}+40$	ns
<b>External Clock</b>							
$t_{CHCX}$	5	High time <sup>4</sup>	20		20		ns
$t_{CLCX}$	5	Low time <sup>4</sup>	20		20		ns
$t_{CLCH}$	5	Rise time <sup>4</sup>		20		20	ns
$t_{CHCL}$	5	Fall time <sup>4</sup>		20		20	ns
<b>Serial Timing – Shift Register Mode<sup>4</sup></b> (Test Conditions: $T_{amb} = 0^{\circ}\text{C}$ to $+70^{\circ}\text{C}$ ; $V_{SS} = 0\text{V}$ ; Load Capacitance = 80pF)							
$t_{XLXL}$	6	Serial port clock cycle time	0.75		$12t_{CLCL}$		$\mu\text{s}$
$t_{QVXH}$	6	Output data setup to clock rising edge	492		$10t_{CLCL}-133$		ns
$t_{XHQX}$	6	Output data hold after clock rising edge	8		$2t_{CLCL}-117$		ns
$t_{XHDX}$	6	Input data hold after clock rising edge	0		0		ns
$t_{XHDV}$	6	Clock rising edge to input data valid		492		$10t_{CLCL}-133$	ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- $t_{CLCL} = 1/f_{OSC}$  = one oscillator clock period.  
 $t_{CLCL} = 83.3\text{ns}$  at  $f_{OSC} = 12\text{MHz}$ .  
 $t_{CLCL} = 62.5\text{ns}$  at  $f_{OSC} = 16\text{MHz}$ .
- These values are characterized but not 100% production tested.

## Single-chip 8-bit microcontroller

80C552/83C552

AC ELECTRICAL CHARACTERISTICS (Continued)<sup>1, 2</sup>

24/30 MHz version

SYMBOL	FIGURE	PARAMETER	24MHz CLOCK		30MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	
1/f <sub>CLCL</sub>	2	Oscillator frequency					1.2	24	MHz
t <sub>LHLL</sub>	2	ALE pulse width	43		27		2t <sub>CLCL</sub> -40		ns
t <sub>AVLL</sub>	2	Address valid to ALE low	17		8		t <sub>CLCL</sub> -25		ns
t <sub>LLAX</sub>	2	Address hold after ALE low	17		8		t <sub>CLCL</sub> -25		ns
t <sub>LLIV</sub>	2	ALE low to valid instruction in		102		68		4t <sub>CLCL</sub> -65	ns
t <sub>LLPL</sub>	2	ALE low to PSEN low	17		8		t <sub>CLCL</sub> -25		ns
t <sub>PLPH</sub>	2	PSEN pulse width	80		55		3t <sub>CLCL</sub> -45		ns
t <sub>PLIV</sub>	2	PSEN low to valid instruction in		65		40		3t <sub>CLCL</sub> -60	ns
t <sub>PXIX</sub>	2	Input instruction hold after PSEN	0		0		0		ns
t <sub>PXIZ</sub>	2	Input instruction float after PSEN		17		8		t <sub>CLCL</sub> -25	ns
t <sub>AVIV</sub>	2	Address to valid instruction in		128		87		5t <sub>CLCL</sub> -80	ns
t <sub>PLAZ</sub>	2	PSEN low to address float		10		10		10	ns
<b>Data Memory</b>									
t <sub>RLRH</sub>	3	RD pulse width	150		100		6t <sub>CLCL</sub> -100		ns
t <sub>WLWH</sub>	4	WR pulse width	150		100		6t <sub>CLCL</sub> -100		ns
t <sub>RLDV</sub>	3	RD low to valid data in		118		77		5t <sub>CLCL</sub> -90	ns
t <sub>RHDZ</sub>	3	Data hold after RD	0		0		0		ns
t <sub>RHDZ</sub>	3	Data float after RD		55		39		2t <sub>CLCL</sub> -28	ns
t <sub>LLDV</sub>	3	ALE low to valid data in		183		117		8t <sub>CLCL</sub> -150	ns
t <sub>AVDV</sub>	3	Address to valid data in		210		135		9t <sub>CLCL</sub> -165	ns
t <sub>LLWL</sub>	3, 4	ALE low to RD or WR low	75	175	50	150	3t <sub>CLCL</sub> -50	3t <sub>CLCL</sub> +50	ns
t <sub>AVWL</sub>	3, 4	Address valid to WR low or RD low	92		58		4t <sub>CLCL</sub> -75		ns
t <sub>QVWX</sub>	4	Data valid to WR transition	12		3		t <sub>CLCL</sub> -30		ns
t <sub>DW</sub>	4	Data before WR	162		103		7t <sub>CLCL</sub> -130		ns
t <sub>WHQX</sub>	4	Data hold after WR	17		8		t <sub>CLCL</sub> -25		ns
t <sub>RLAZ</sub>	3	RD low to address float		0		0		0	ns
t <sub>WHLH</sub>	3, 4	RD or WR high to ALE high	17	67	8	58	t <sub>CLCL</sub> -25	t <sub>CLCL</sub> +25	ns
<b>External Clock</b>									
t <sub>CHCX</sub>	5	High time <sup>3</sup>	17		15		17		ns
t <sub>CLCX</sub>	5	Low time <sup>3</sup>	17		15		17		ns
t <sub>CLCH</sub>	5	Rise time <sup>3</sup>		5		3		20	ns
t <sub>CHCL</sub>	5	Fall time <sup>3</sup>		5		3		20	ns
<b>Serial Timing – Shift Register Mode<sup>3</sup></b> (Test Conditions: T <sub>amb</sub> = 0°C to +70°C; V <sub>SS</sub> = 0V; Load Capacitance = 80pF)									
t <sub>XLXL</sub>	6	Serial port clock cycle time	0.5		0.4		12t <sub>CLCL</sub>		μs
t <sub>QVXH</sub>	6	Output data setup to clock rising edge	283		200		10t <sub>CLCL</sub> -133		ns
t <sub>XHQX</sub>	6	Output data hold after clock rising edge	23		6.6		2t <sub>CLCL</sub> -60		ns
t <sub>XHDX</sub>	6	Input data hold after clock rising edge	0		0		0		ns
t <sub>XHDV</sub>	6	Clock rising edge to input data valid		283		200		10t <sub>CLCL</sub> -133	ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- These values are characterized but not 100% production tested.
- t<sub>CLCL</sub> = 1/f<sub>OSC</sub> = one oscillator clock period.  
t<sub>CLCL</sub> = 41.7ns at f<sub>OSC</sub> = 24MHz.

## Single-chip 8-bit microcontroller

80C552/83C552

## AC ELECTRICAL CHARACTERISTICS (Continued)

SYMBOL	PARAMETER	INPUT	OUTPUT
<b>I<sup>2</sup>C Interface (Refer to Figure 9)</b>			
t <sub>HD</sub> : STA	START condition hold time	≥ 14 t <sub>CLCL</sub>	> 4.0 μs <sup>1</sup>
t <sub>LOW</sub>	SCL low time	≥ 16 t <sub>CLCL</sub>	> 4.7 μs <sup>1</sup>
t <sub>HIGH</sub>	SCL high time	≥ 14 t <sub>CLCL</sub>	> 4.0 μs <sup>1</sup>
t <sub>RC</sub>	SCL rise time	≤ 1 μs	- <sup>2</sup>
t <sub>FC</sub>	SCL fall time	≤ 0.3 μs	< 0.3 μs <sup>3</sup>
t <sub>SU</sub> : DAT1	Data set-up time	≥ 250 ns	> 20 t <sub>CLCL</sub> - t <sub>RD</sub>
t <sub>SU</sub> : DAT2	SDA set-up time (before rep. START cond.)	≥ 250 ns	> 1 μs <sup>1</sup>
t <sub>SU</sub> : DAT3	SDA set-up time (before STOP cond.)	≥ 250 ns	> 8 t <sub>CLCL</sub>
t <sub>HD</sub> : DAT	Data hold time	≥ 0 ns	> 8 t <sub>CLCL</sub> - t <sub>FC</sub>
t <sub>SU</sub> : STA	Repeated START set-up time	≥ 14 t <sub>CLCL</sub>	> 4.7 μs <sup>1</sup>
t <sub>SU</sub> : STO	STOP condition set-up time	≥ 14 t <sub>CLCL</sub>	> 4.0 μs <sup>1</sup>
t <sub>BUF</sub>	Bus free time	≥ 14 t <sub>CLCL</sub>	> 4.7 μs <sup>1</sup>
t <sub>RD</sub>	SDA rise time	≤ 1 μs	- <sup>2</sup>
t <sub>FD</sub>	SDA fall time	≤ 0.3 μs	< 0.3 μs <sup>3</sup>

## NOTES:

- At 100 kbit/s. At other bit rates this value is inversely proportional to the bit-rate of 100 kbit/s.
- Determined by the external bus-line capacitance and the external bus-line pull-resistor, this must be < 1 μs.
- Spikes on the SDA and SCL lines with a duration of less than 3 t<sub>CLCL</sub> will be filtered out. Maximum capacitance on bus-lines SDA and SCL = 400 pF.
- t<sub>CLCL</sub> = 1/f<sub>OSC</sub> = one oscillator clock period at pin XTAL1. For 62 ns, 42 ns, 33.3 ns < t<sub>CLCL</sub> < 285 ns (16 MHz, 24 MHz, 30 MHz > f<sub>OSC</sub> > 1.2 MHz) the Si01 interface meets the I<sup>2</sup>C-bus specification for bit-rates up to 100 kbit/s.

ns	80	125	210	ns	Addresses to valid data in
ns	50	80	125	ns	ALE low to RD or WR low
ns	25	50	95	ns	Addresses valid to WR low or RD low
ns	10	3	15	ns	Data valid to WR transition
ns	150	100	185	ns	Data before WR
ns	25	8	17	ns	Data hold after WR
ns	0	0	0	ns	RD low to address float
ns	100 + 25	88	87	ns	RD or WR high to ALE high
External Clock					
ns	17	15	17	ns	High time
ns	17	15	17	ns	Low time
ns	20	3	5	ns	Rise time
ns	20	3	5	ns	Fall time
Signal Timing - Shift Register Mode (Test Conditions: T <sub>amb</sub> = 0°C to +70°C; V <sub>DD</sub> = 0V; Load Capacitance = 50pF)					
ns	120	0.4	0.3	ns	Signal non clock cycle time
ns	100 + 125	200	288	ns	Output data setup to clock rising edge
ns	200 + 80	8.8	33	ns	Output data hold after clock rising edge
ns	0	0	0	ns	Input data hold after clock rising edge
ns	100 + 125	200	288	ns	Clock rising edge to input data valid

NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100 pF; load capacitance for all other outputs = 50 pF.
- These values are characterized but not 100% production tested.
- f<sub>OSC</sub> = 1/f<sub>OSC</sub> = one oscillator clock period.
- f<sub>OSC</sub> = 41.1 MHz at f<sub>OSC</sub> = 24 MHz.

## Single-chip 8-bit microcontroller

80C552/83C552

## EXPLANATION OF THE AC SYMBOLS

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

A – Address  
C – Clock  
D – Input data  
H – Logic level high  
I – Instruction (program memory contents)  
L – Logic level low, or ALE  
P – PSEN

Q – Output data

R – RD signal

t – Time

V – Valid

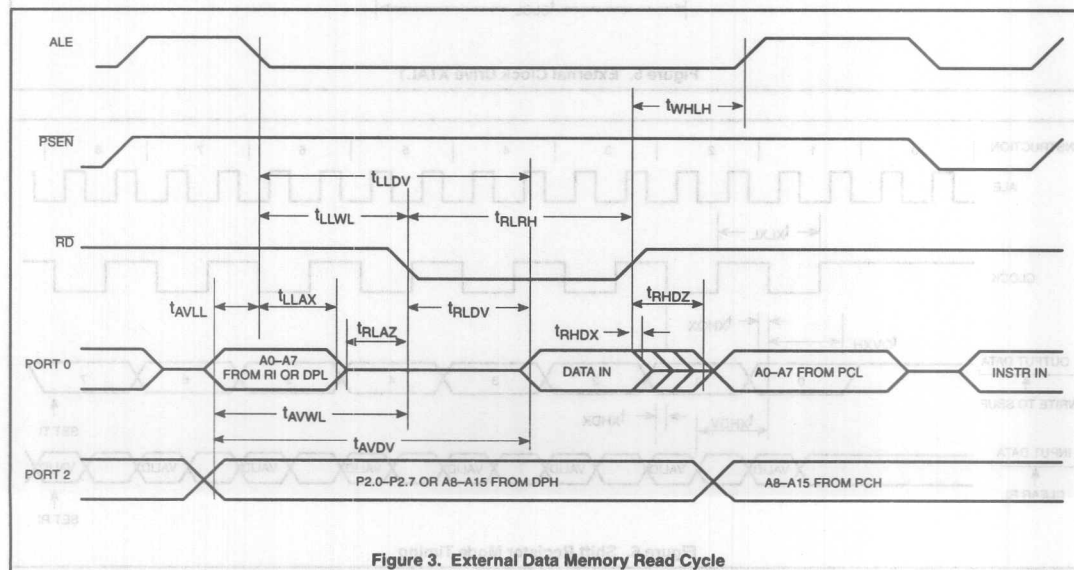
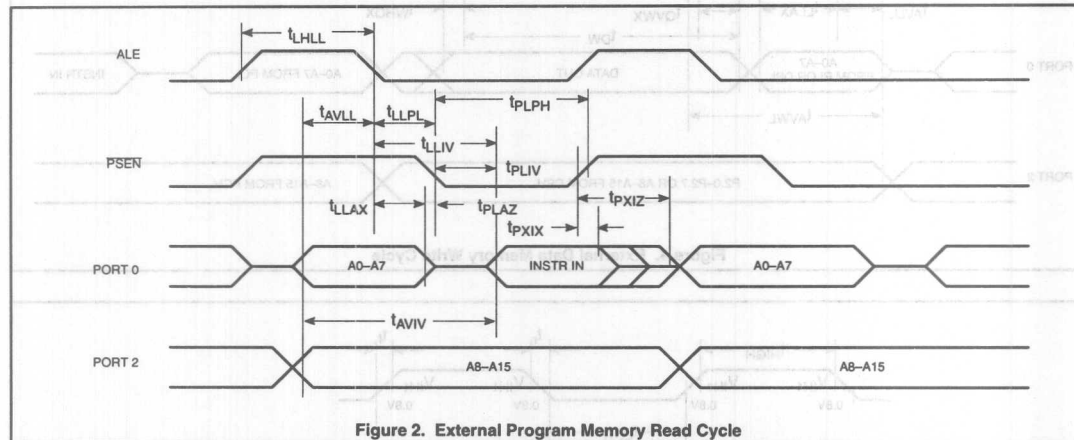
W – WR signal

X – No longer a valid logic level

Z – Float

Examples:  $t_{AVLL}$  = Time for address valid to ALE low.

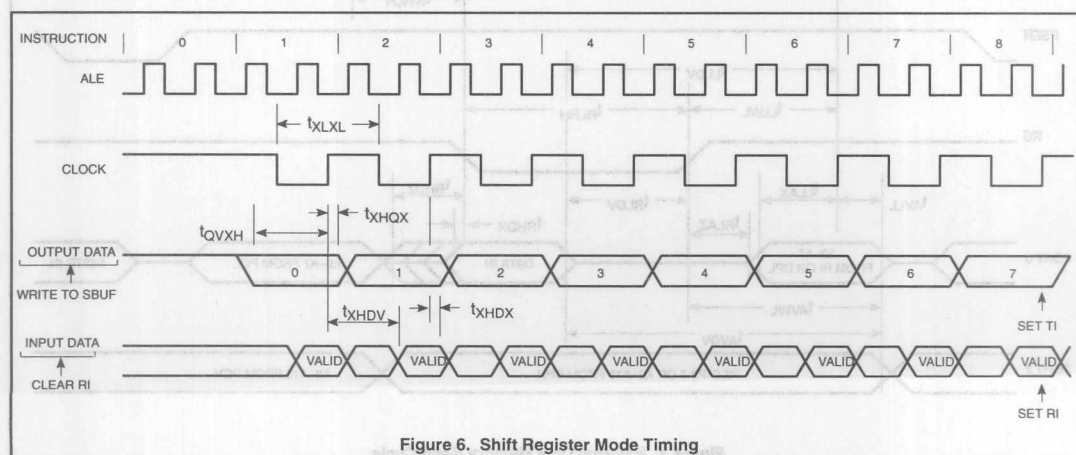
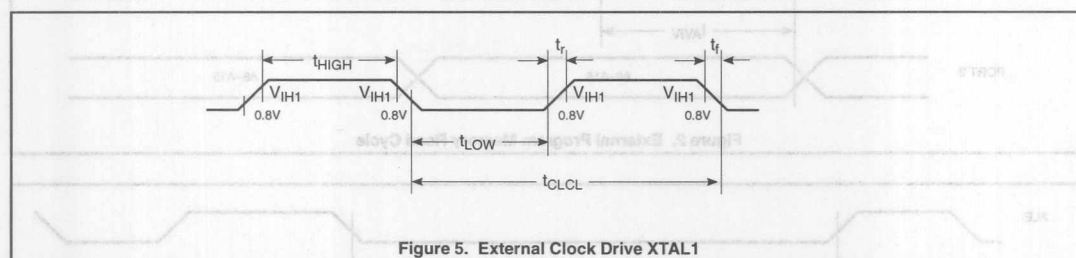
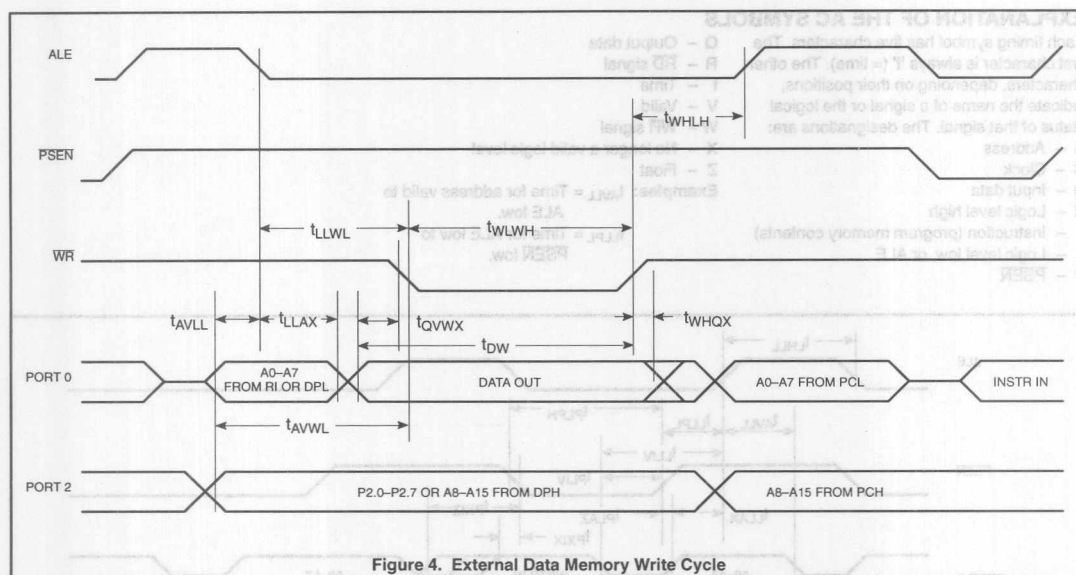
$t_{LLPL}$  = Time for ALE low to PSEN low.





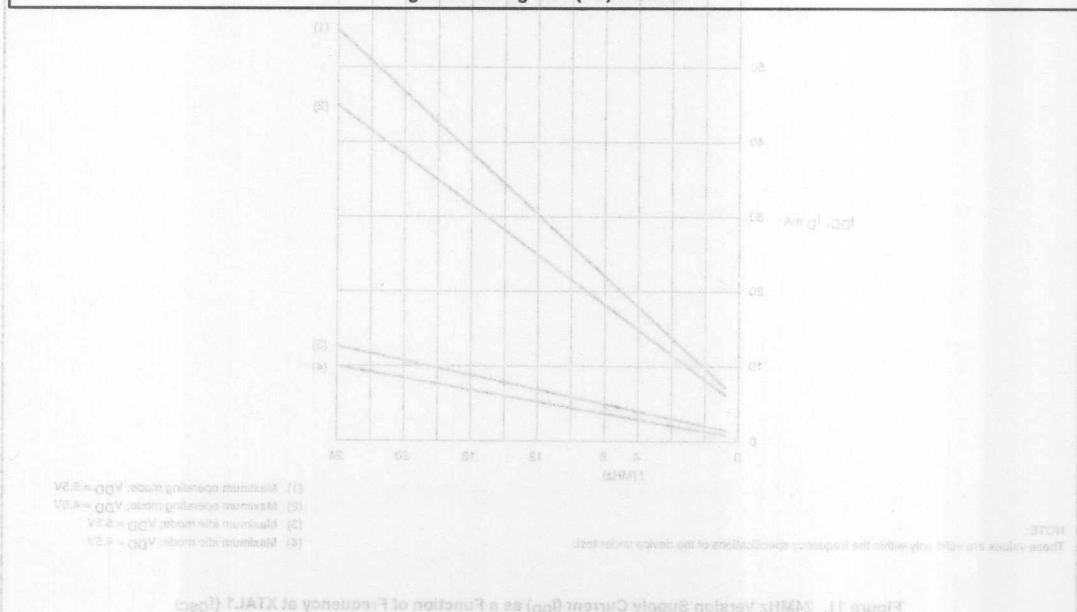
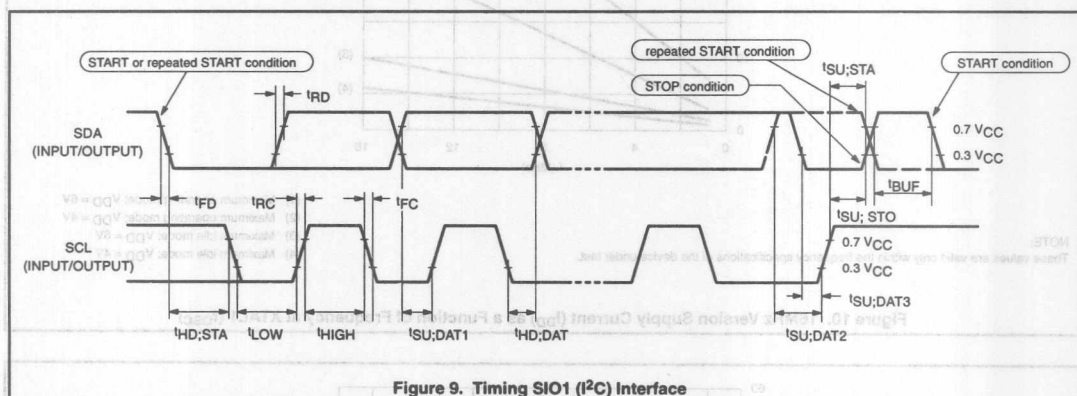
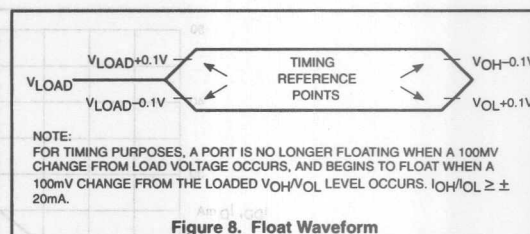
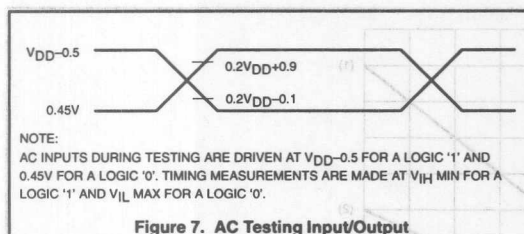
## Single-chip 8-bit microcontroller

80C552/83C552



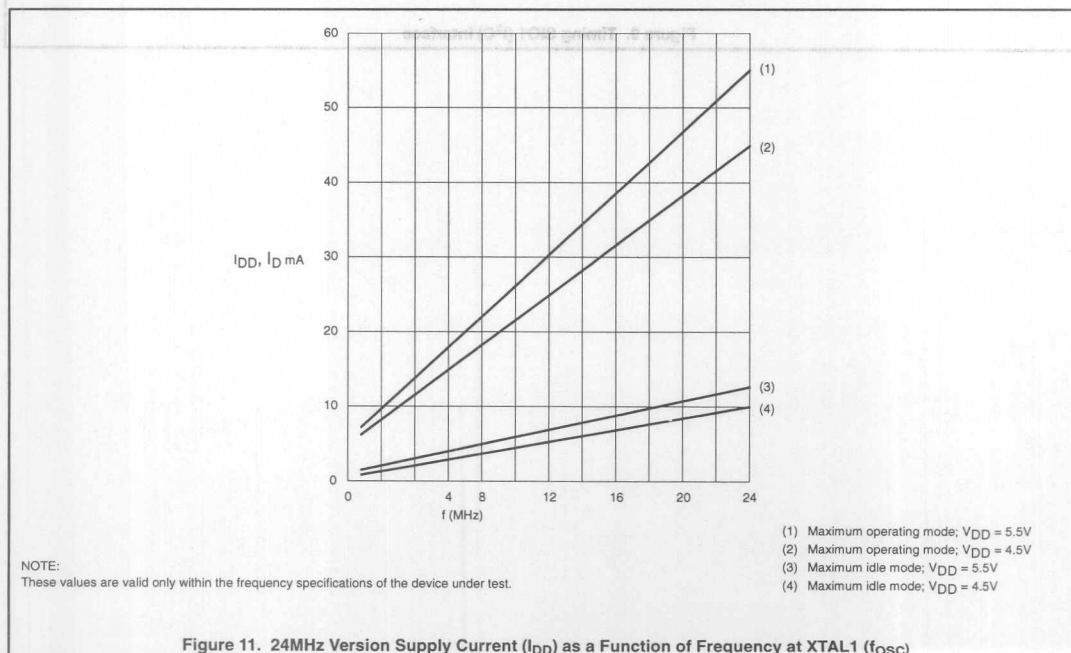
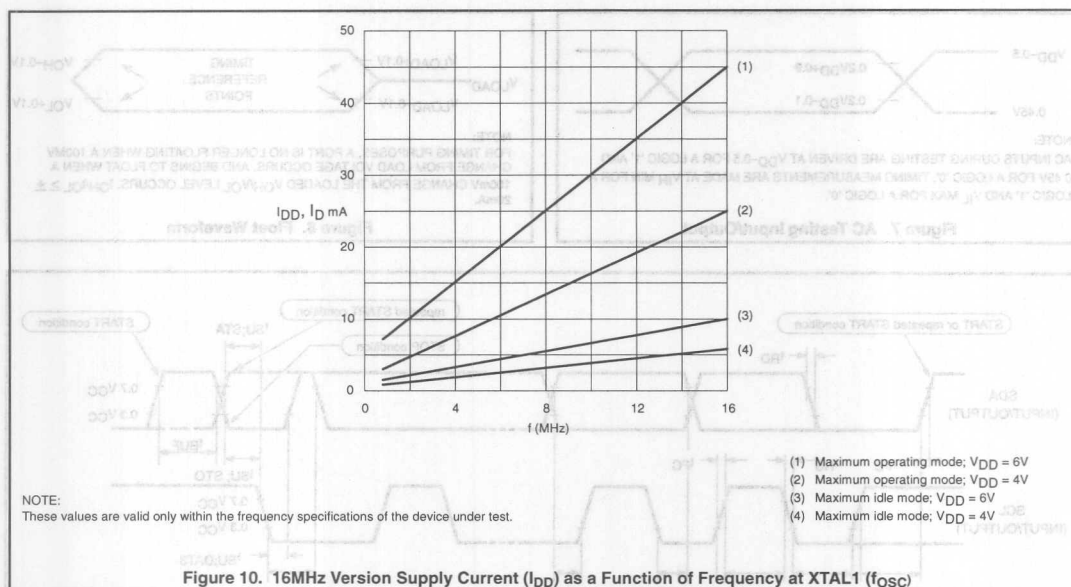
# Single-chip 8-bit microcontroller

80C552/83C552



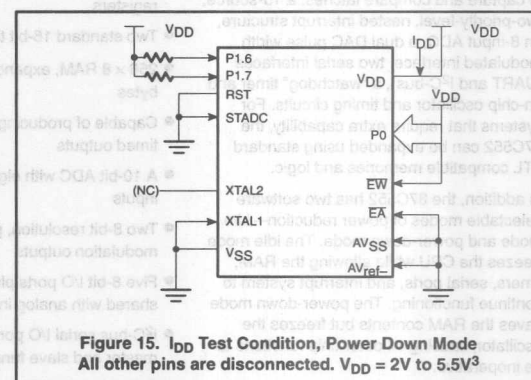
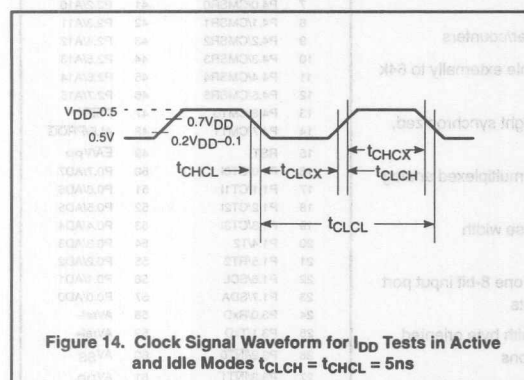
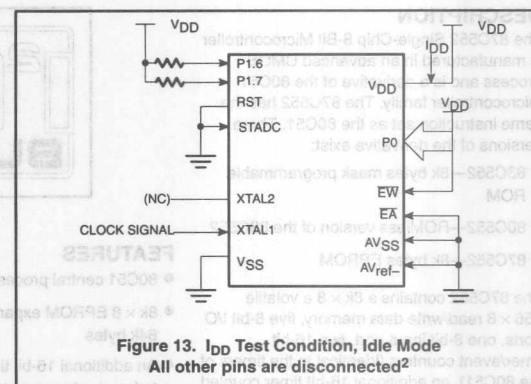
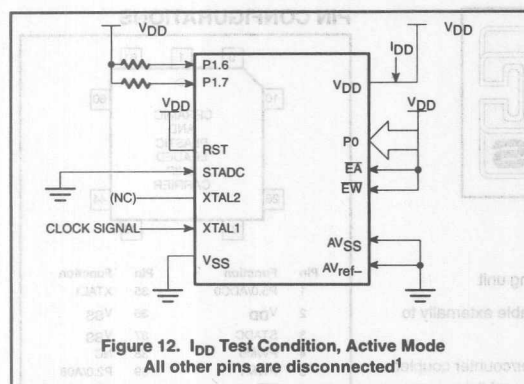
## Single-chip 8-bit microcontroller

## 80C552/83C552



## Single-chip 8-bit microcontroller

## 80C552/83C552



## NOTES:

- Active Mode:
  - The following pins must be forced to  $V_{DD}$ :  $\overline{EA}$ , RST, Port 0, and EW.
  - The following pins must be forced to  $V_{SS}$ : STADC,  $AV_{SS}$ , and  $AV_{ref-}$ .
  - Ports 1.6 and 1.7 should be connected to  $V_{DD}$  through resistors of sufficiently high value such that the sink current into these pins cannot exceed the  $I_{OL1}$  spec of these pins.
  - The following pins must be disconnected: XTAL2 and all pins not specified above.
- Idle Mode:
  - The following pins must be forced to  $V_{DD}$ : Port 0 and EW.
  - The following pins must be forced to  $V_{SS}$ : RST, STADC,  $AV_{SS}$ ,  $AV_{ref-}$ , and  $\overline{EA}$ .
  - Ports 1.6 and 1.7 should be connected to  $V_{DD}$  through resistors of sufficiently high value such that the sink current into these pins cannot exceed the  $I_{OL1}$  spec of these pins. These pins must not have logic 0 written to them prior to this measurement.
  - The following pins must be disconnected: XTAL2 and all pins not specified above.
- Power Down Mode:
  - The following pins must be forced to  $V_{DD}$ : Port 0 and EW.
  - The following pins must be forced to  $V_{SS}$ : RST, STADC, XTAL1,  $AV_{SS}$ ,  $AV_{ref-}$ , and  $\overline{EA}$ .
  - Ports 1.6 and 1.7 should be connected to  $V_{DD}$  through resistors of sufficiently high value such that the sink current into these pins cannot exceed the  $I_{OL1}$  spec of these pins. These pins must not have logic 0 written to them prior to this measurement.
  - The following pins must be disconnected: XTAL2 and all pins not specified above.



Purchase of Philips' I<sup>2</sup>C components conveys a license under the Philips' I<sup>2</sup>C patent to use the components in the I<sup>2</sup>C-system provided the system conforms to the I<sup>2</sup>C specifications defined by Philips.

## Single-chip 8-bit microcontroller

87C552

Single-chip 8-bit microcontroller with 10-bit A/D, capture/compare timer, high-speed outputs, PWM

## DESCRIPTION

The 87C552 Single-Chip 8-Bit Microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 87C552 has the same instruction set as the 80C51. Three versions of the derivative exist:

- 83C552—8k bytes mask programmable ROM
- 80C552—ROMless version of the 83C552
- 87C552—8k bytes EPROM

The 87C552 contains a 8k × 8 volatile 256 × 8 read/write data memory, five 8-bit I/O ports, one 8-bit input port, two 16-bit timer/event counters (identical to the timers of the 80C51), an additional 16-bit timer coupled to capture and compare latches, a 15-source, two-priority-level, nested interrupt structure, an 8-input ADC, a dual DAC pulse width modulated interface, two serial interfaces (UART and I<sup>2</sup>C-bus), a "watchdog" timer and on-chip oscillator and timing circuits. For systems that require extra capability, the 87C552 can be expanded using standard TTL compatible memories and logic.

In addition, the 87C552 has two software selectable modes of power reduction—idle mode and power-down mode. The idle mode freezes the CPU while allowing the RAM, timers, serial ports, and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.

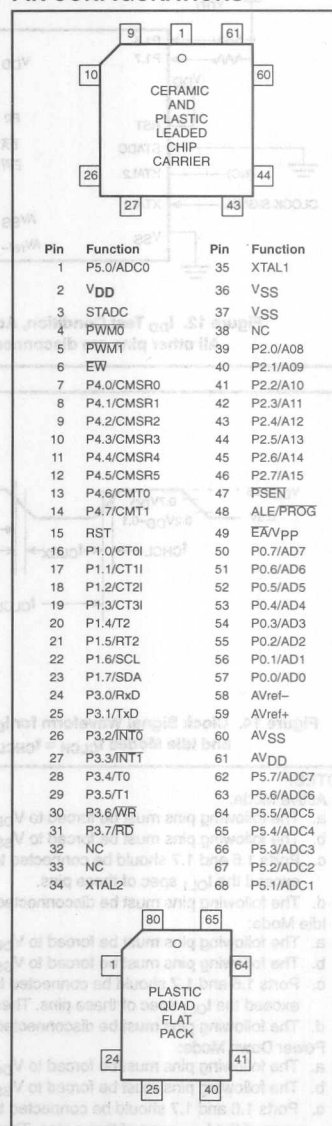
The device also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of over 100 instructions: 49 one-byte, 45 two-byte, and 17 three-byte. With a 16MHz (24MHz) crystal, 58% of the instructions are executed in 0.75μs (0.5μs) and 40% in 1.5μs (1μs). Multiply and divide instructions require 3μs (2μs).



## FEATURES

- 80C51 central processing unit
- 8k × 8 EPROM expandable externally to 64k bytes
- An additional 16-bit timer/counter coupled to four capture registers and three compare registers
- Two standard 16-bit timer/counters
- 256 × 8 RAM, expandable externally to 64k bytes
- Capable of producing eight synchronized, timed outputs
- A 10-bit ADC with eight multiplexed analog inputs
- Two 8-bit resolution, pulse width modulation outputs
- Five 8-bit I/O ports plus one 8-bit input port shared with analog inputs
- I<sup>2</sup>C-bus serial I/O port with byte oriented master and slave functions
- Full-duplex UART compatible with the standard 80C51
- On-chip watchdog timer
- Three speed ranges:
  - 16MHz
- Extended temperature ranges
- OTP package available

## PIN CONFIGURATIONS



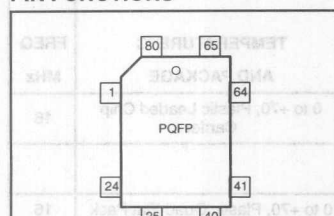


## Single-chip 8-bit microcontroller

87C552

## PLASTIC QUAD FLAT PACK

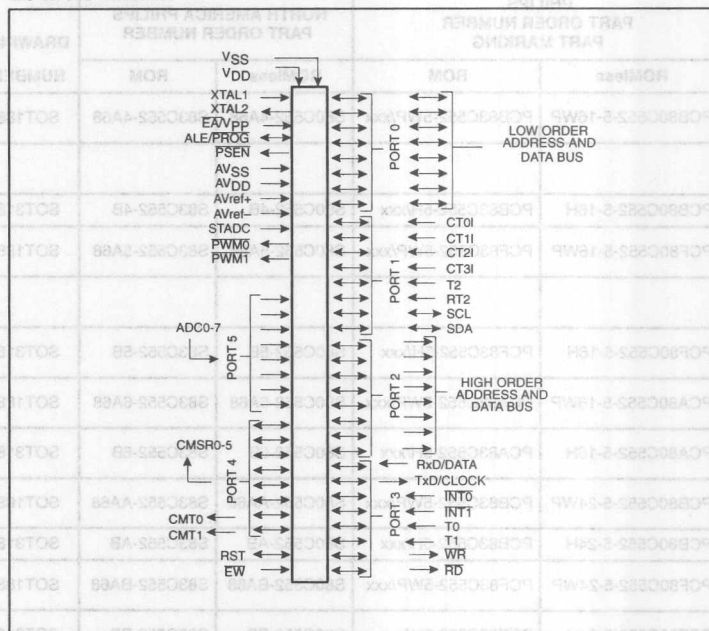
## PIN FUNCTIONS



Pin	Function	Pin	Function
1	P4.1/CMSR1	41	P2.3/A11
2	P4.2/CMSR2	42	P2.4/A12
3	NC	43	NC
4	P4.3/CMSR3	44	NC
5	P4.4/CMSR4	45	P2.5/A13
6	P4.5/CMSR5	46	P2.6/A14
7	P4.6/CMT0	47	P2.7/A15
8	P4.7/CMT1	48	PSEN
9	RST	49	ALE/PROG
10	P1.0/CT0I	50	EA/Vpp
11	P1.1/CT1I	51	P0.7/AD7
12	P1.2/CT2I	52	P0.6/AD6
13	P1.3/CT3I	53	P0.5/AD5
14	P1.4/T2	54	P0.4/AD4
15	P1.5/RT2	55	P0.3/AD3
16	P1.6/SCL	56	P0.2/AD2
17	P1.7/SDA	57	P0.1/AD1
18	P3.0/RxD	58	P0.0/AD0
19	P3.1/TxD	59	AVref-
20	P3.2/INT0	60	AVref+
21	NC	61	AVSS
22	NC	62	NC
23	P3.3/INT1	63	AVDD
24	P3.4/T0	64	P5.7/ADC7
25	P3.5/T1	65	P5.6/ADC6
26	P3.6/WR	66	P5.5/ADC5
27	P3.7/RD	67	P5.4/ADC4
28	NC	68	P5.3/ADC3
29	NC	69	P5.2/ADC2
30	NC	70	P5.1/ADC1
31	XTAL2	71	P5.0/ADC0
32	XTAL1	72	VDD
33	IC	73	IC
34	VSS	74	STADC
35	VSS	75	PWM0
36	VSS	76	PWM1
37	NC	77	EW
38	P2.0/A08	78	NC
39	P2.1/A09	79	NC
40	P2.2/A10	80	P4.0/CMSR0

NC = not connected  
IC = internally connected (do not use)

## LOGIC SYMBOL



## Single-chip 8-bit microcontroller

## 87C552

## ORDERING INFORMATION

PHILIPS PART ORDER NUMBER PART MARKING		NORTH AMERICA PHILIPS PART ORDER NUMBER		DRAWING  NUMBER	TEMPERATURE °C  AND PACKAGE	FREQ  MHz
ROMless	ROM	ROMless	ROM			
PCB80C552-5-16WP	PCB83C552-5WP/xxx	S80C552-4A68	S83C552-4A68	SOT188	0 to +70, Plastic Leaded Chip Carrier	16
PCB80C552-5-16H	PCB83C552-5H/xxx	S80C552-4B	S83C552-4B	SOT318	0 to +70, Plastic Quad Flat Pack	16
PCF80C552-5-16WP	PCF83C552-5WP/xxx	S80C552-5A68	S83C552-5A68	SOT188	-40 to +85, Plastic Leaded Chip Carrier	16
PCF80C552-5-16H	PCF83C552-5H/xxx	S80C552-5B	S83C552-5B	SOT318	-40 to +85, Plastic Quad Flat Pack	16
PCA80C552-5-16WP	PCA83C552-5WP/xxx	S80C552-6A68	S83C552-6A68	SOT188	-40 to +125, Plastic Leaded Chip Carrier	16
PCA80C552-5-16H	PCA83C552-5H/xxx	S80C552-6B	S83C552-6B	SOT318	-40 to +125, Plastic Quad Flat Pack	16
PCB80C552-5-24WP	PCB83C552-5WP/xxx	S80C552-AA68	S83C552-AA68	SOT188	0 to +70, Plastic Leaded Chip Carrier	24
PCB80C552-5-24H	PCB83C552-5H/xxx	S80C552-AB	S83C552-AB	SOT318	0 to +70, Plastic Quad Flat Pack	24
PCF80C552-5-24WP	PCF83C552-5WP/xxx	S80C552-BA68	S83C552-BA68	SOT188	-40 to +85, Plastic Leaded Chip Carrier	24
PCF80C552-5-24H	PCF83C552-5H/xxx	S80C552-BB	S83C552-BB	SOT318	-40 to +85, Plastic Quad Flat Pack	24
PCB80C552-5-30WP	PCB83C552-5WP/xxx	S80C552-CA68	S83C552-CA68	SOT188	0 to +70, Plastic Leaded Chip Carrier	30
PCB80C552-5-30H	PCB83C552-5H/xxx	S80C552-CB	S83C552-CB	SOT318	0 to +70, Plastic Quad Flat Pack	30

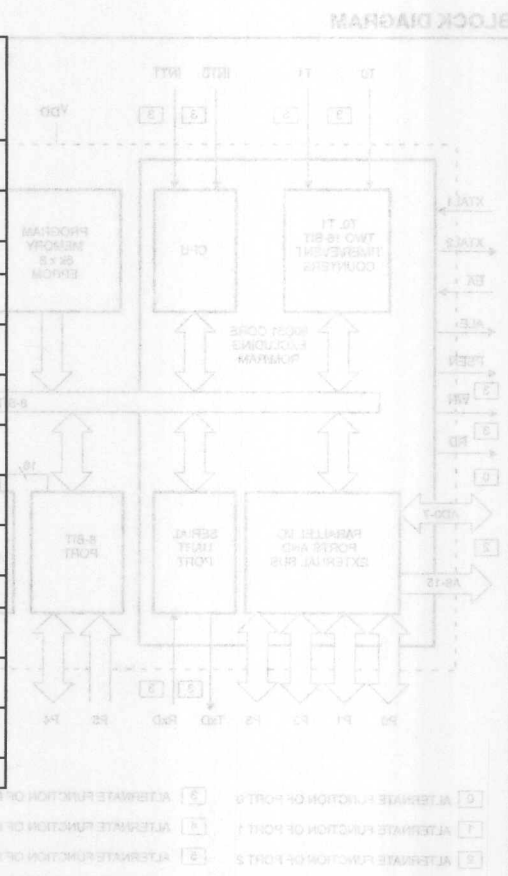
**NOTE:**

1. xxx denotes the ROM code number.

Year	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100
1990	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100

## Single-chip 8-bit microcontroller

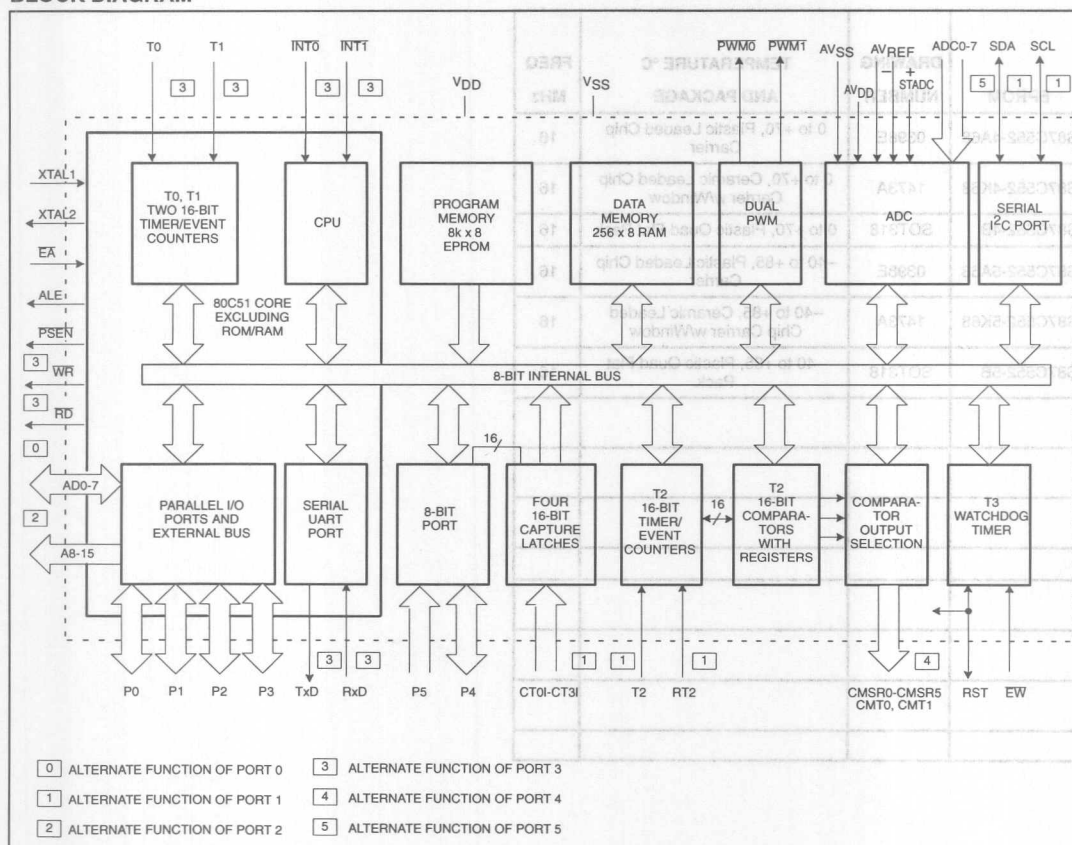
## 87C552

[illegible]

## Single-chip 8-bit microcontroller

## 87C552

## BLOCK DIAGRAM



## Single-chip 8-bit microcontroller

87C552

## PIN DESCRIPTION

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	PLCC	QFP		
V <sub>DD</sub>	2	72	I	<b>Digital Power Supply:</b> +5V power supply pin during normal operation, idle and power-down mode.
STADC	3	74	I	<b>Start ADC Operation:</b> Input starting analog to digital conversion (ADC operation can also be started by software).
PWM0	4	75	O	<b>Pulse Width Modulation:</b> Output 0.
PWM1	5	76	O	<b>Pulse Width Modulation:</b> Output 1.
EW	6	77	I	<b>Enable Watchdog Timer:</b> Enable for T3 watchdog timer and disable power-down mode.
P0.0-P0.7	57-50	58-51	I/O	<b>Port 0:</b> Port 0 is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application it uses strong internal pull-ups when emitting 1s. Port 0 is also used to input the code byte during programming and to output the code byte during verification.
P1.0-P1.7	16-23	10-17	I/O	<b>Port 1:</b> 8-bit I/O port. Alternate functions include:
	16-21	10-15	I/O	(P1.0-P1.5): Quasi-bidirectional port pins.
	22-23	16-17	I/O	(P1.6, P1.7): Open drain port pins.
	16-19	10-13	I	CT0I-CT3I (P1.0-P1.3): Capture timer input signals for timer T2.
	20	14	I	T2 (P1.4): T2 event input.
	21	15	I	RT2 (P1.5): T2 timer reset signal. Rising edge triggered.
	22	16	I/O	SCL (P1.6): Serial port clock line I <sup>2</sup> C-bus.
P2.0-P2.7	23	17	I/O	SDA (P1.7): Serial port data line I <sup>2</sup> C-bus.
				Port 1 is also used to input the lower order address byte during EPROM programming and verification. A0 is on P1.0, etc.
	39-46	38-42, 45-47	I/O	<b>Port 2:</b> 8-bit quasi-bidirectional I/O port.
				Alternate function: High-order address byte for external memory (A08-A15). Port 2 is also used to input the upper order address during EPROM programming and verification. A8 is on P2.0, A9 on P2.1, through A12 on P2.4.
	24-31	18-20, 23-27	I/O	<b>Port 3:</b> 8-bit quasi-bidirectional I/O port. Alternate functions include:
	24	18		RxD (P3.0): Serial input port.
	25	19		TxD (P3.1): Serial output port.
	26	20		INT0 (P3.2): External interrupt.
	27	23		INT1 (P3.3): External interrupt.
	28	24		T0 (P3.4): Timer 0 external input.
P4.0-P4.7	29	25		T1 (P3.5): Timer 1 external input.
	30	26		WR (P3.6): External data memory write strobe.
	31	27		RD (P3.7): External data memory read strobe.
	7-14	80, 1-2, 4-8	I/O	<b>Port 4:</b> 8-bit quasi-bidirectional I/O port. Alternate functions include:
	7-12	80, 1-2, 4-6	O	CMSR0-CMSR5 (P4.0-P4.5): Timer T2 compare and set/reset outputs on a match with timer T2.
P5.0-P5.7	13, 14	7, 8	O	CMT0, CMT1 (P4.6, P4.7): Timer T2 compare and toggle outputs on a match with timer T2.
	68-62, 1	71-64, 1	I	<b>Port 5:</b> 8-bit input port.
RST	15	9	I/O	<b>ADC0-ADC7 (P5.0-P5.7):</b> Alternate function: Eight input channels to ADC.
				<b>Reset:</b> Input to reset the 87C552. It also provides a reset pulse as output when timer T3 overflows.
XTAL1	35	32	I	<b>Crystal Input 1:</b> Input to the inverting amplifier that forms the oscillator, and input to the internal clock generator. Receives the external clock signal when an external oscillator is used.
XTAL2	34	31	O	<b>Crystal Input 2:</b> Output of the inverting amplifier that forms the oscillator. Left open-circuit when an external clock is used.
V <sub>SS</sub>	36, 37	34-36	I	<b>Digital ground.</b>
PSEN	47	48	O	<b>Program Store Enable:</b> Active-low read strobe to external program memory.



Single-chip 8-bit microcontroller

87C552

PIN DESCRIPTION (Continued)

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	PLCC	QFP		
ALE/PROG	48	49	O	<b>Address Latch Enable:</b> Latches the low byte of the address during accesses to external memory. It is activated every six oscillator periods. During an external data memory access, one ALE pulse is skipped. ALE can drive up to eight LS TTL inputs and handles CMOS inputs without an external pull-up. This pin is also the program pulse input (PROG) during EPROM programming.
EA/V <sub>PP</sub>	49	50	I	<b>External Access:</b> When EA is held at TTL level high, the CPU executes out of the internal program ROM provided the program counter is less than 8192. When EA is held at TTL low level, the CPU executes out of external program memory. EA is not allowed to float. This pin also receives the 12.75V programming supply voltage (V <sub>PP</sub> ) during EPROM programming.
AV <sub>REF-</sub>	58	59	I	<b>Analog to Digital Conversion Reference Resistor:</b> Low-end.
AV <sub>REF+</sub>	59	60	I	<b>Analog to Digital Conversion Reference Resistor:</b> High-end.
AV <sub>SS</sub>	60	61	I	<b>Analog Ground</b>
AV <sub>DD</sub>	61	63	I	<b>Analog Power Supply</b>

NOTE:

1. To avoid "latch-up" effect at power-on, the voltage on any pin at any time must not be higher or lower than V<sub>DD</sub> + 0.5V or V<sub>SS</sub> - 0.5V, respectively.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator, as shown in the logic symbol.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

RESET

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. To insure a good power-on reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few milliseconds) plus two machine cycles. At power-on, the voltage on V<sub>DD</sub> and RST must come up at the same time for a proper start-up.

IDLE MODE

In the idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers

remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

POWER-DOWN MODE

In the power-down mode, the oscillator is stopped and the instruction to invoke power-down is the last instruction executed. Only the contents of the on-chip RAM are preserved. A hardware reset is the only way to terminate the power-down mode. The control bits for the reduced power modes are in the special function register PCON. Table 1 shows the state of the I/O ports during low current operating modes.

Table 1. External Pin Status During Idle and Power-Down Modes

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3	PORT 4	PWM0/ PWM1
Idle	Internal	1	1	Data	Data	Data	Data	Data	High
Idle	External	1	1	Float	Data	Address	Data	Data	High
Power-down	Internal	0	0	Data	Data	Data	Data	Data	High
Power-down	External	0	0	Float	Data	Data	Data	Data	High

## Single-chip 8-bit microcontroller

87C552

## Serial Control Register (S1CON) – See Table 2

S1CON (D8H)	CR2	ENS1	STA	STO	SI	AA	CR1	CR0
-------------	-----	------	-----	-----	----	----	-----	-----

Bits CR0, CR1 and CR2 determine the serial clock frequency that is generated in the master mode of operation.

Table 2. Serial Clock Rates

CR2	CR1	CR0	BIT FREQUENCY (kHz) AT $f_{osc}$			$f_{osc}$ DIVIDED BY
			6MHz	12MHz	16MHz	
0	0	0	23	47	62.5	256
0	0	1	27	54	71	224
0	1	0	31.25	62.5	83.3	192
0	1	1	37	75	100	160
1	0	0	6.25	12.5	17	960
1	0	1	50	100	133	120
1	1	0	100	200	267	60
1	1	1	0.25 < 62.5 0 to 225	0.5 < 62.5 0 to 224	0.67 < 56 0 to 223	96 × (256 – (reload value Timer 1)) Timer 1 in Mode 2.

## NOTE:

1. These frequencies exceed the upper limit of 100kHz of the I<sup>2</sup>C-bus specification and cannot be used in an I<sup>2</sup>C-bus application.

ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>

PARAMETER	RATING	UNIT
Storage temperature range	–65 to +150	°C
Voltage on EA/V <sub>PP</sub> to V <sub>SS</sub>	–0.5 to +13	V
Voltage on any other pin to V <sub>SS</sub>	–0.5 to +6.5	V
Input, output DC current on any single I/O pin	5.0	mA
Power dissipation (based on package heat transfer limitations, not device power consumption)	1.0	W

## NOTES:

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
- This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
- Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V<sub>SS</sub> unless otherwise noted.

## DEVICE SPECIFICATIONS

TYPE	SUPPLY VOLTAGE (V)		FREQUENCY (MHz)		TEMPERATURE RANGE (°C)
	MIN	MAX	MIN	MAX	
P87C552-4	4.5	5.5	3.5	16	0 to +70
P87C552-5	4.5	5.5	3.5	16	–40 to +85

## DC ELECTRICAL CHARACTERISTICS

$V_{SS}, AV_{SS} = 0V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			MIN	MAX	
$I_{DD}$	Supply current operating: PCA8XC552-5-16	See notes 1 and 2 $f_{OSC} = 16MHz$		40	mA
$I_{ID}$	Idle mode: 87C552	See notes 1 and 3 $f_{OSC} = 16MHz$		7	mA
$I_{PD}$	Power-down current: 87C552	See notes 1 and 4; $2V < V_{PD} < V_{DD} \text{ max}$		50	$\mu A$
<b>Inputs</b>					
$V_{IL}$	Input low voltage, except $E\bar{A}$ , P1.6, P1.7		-0.5	$0.2V_{DD}-0.1$	V
$V_{IL1}$	Input low voltage to $E\bar{A}$		-0.5	$0.2V_{DD}-0.3$	V
$V_{IL2}$	Input low voltage to P1.6/SCL, P1.7/SDA <sup>5</sup>		-0.5	$0.3V_{DD}$	V
$V_{IH}$	Input high voltage, except XTAL1, RST		$0.2V_{DD}+0.9$	$V_{DD}+0.5$	V
$V_{IH1}$	Input high voltage, XTAL1, RST		$0.7V_{DD}$	$V_{DD}+0.5$	V
$V_{IH2}$	Input high voltage, P1.6/SCL, P1.7/SDA <sup>5</sup>		$0.7V_{DD}$	6.0	V
$I_{IL}$	Logical 0 input current, ports 1, 2, 3, 4, except P1.6, P1.7	$V_{IN} = 0.45V$		-50	$\mu A$
$I_{TL}$	Logical 1-to-0 transition current, ports 1, 2, 3, 4, except P1.6, P1.7	See note 6		-650	$\mu A$
$\pm I_{IL1}$	Input leakage current, port 0, $E\bar{A}$ , STADC, EW	$0.45V < V_I < V_{DD}$		10	$\mu A$
$\pm I_{IL2}$	Input leakage current, P1.6/SCL, P1.7/SDA	$0V < V_I < 6V$ $0V < V_{DD} < 5.5V$		10	$\mu A$
$\pm I_{IL3}$	Input leakage current, port 5	$0.45V < V_I < V_{DD}$		1	$\mu A$
<b>Outputs</b>					
$V_{OL}$	Output low voltage, ports 1, 2, 3, 4, except P1.6, P1.7	$I_{OL} = 1.6mA^7$		0.45	V
$V_{OL1}$	Output low voltage, port 0, ALE, PSEN, PWM0, PWM1	$I_{OL} = 3.2mA^7$		0.45	V
$V_{OL2}$	Output low voltage, P1.6/SCL, P1.7/SDA	$I_{OL} = 3.0mA^7$		0.4	V
$V_{OH}$	Output high voltage, ports 1, 2, 3, 4, except P1.6/SCL, P1.7/SDA	$-I_{OH} = 60\mu A$ $-I_{OH} = 25\mu A$ $-I_{OH} = 10\mu A$	2.4 $0.75V_{DD}$ $0.9V_{DD}$		V V V
$V_{OH1}$	Output high voltage (port 0 in external bus mode, ALE, PSEN, PWM0, PWM1) <sup>8</sup>	$-I_{OH} = 400\mu A$ $-I_{OH} = 150\mu A$ $-I_{OH} = 40\mu A$	2.4 $0.75V_{DD}$ $0.9V_{DD}$		V V V
$V_{OH2}$	Output high voltage (RST)	$-I_{OH} = 400\mu A$ $-I_{OH} = 120\mu A$	2.4 $0.8V_{DD}$		V V
$R_{RST}$	Internal reset pull-down resistor		50	150	k $\Omega$
$C_{IO}$	Pin capacitance	Test freq = 1MHz, $T_{amb} = 25^\circ C$		10	pF
<b>Analog Inputs</b>					
$AV_{DD}$	Analog supply voltage: 87C552 <sup>9</sup>	$AV_{DD} = V_{DD} \pm 0.2V$	4.5	5.5	V
$AI_{DD}$	Analog supply current: operating:	Port 5 = 0 to $AV_{DD}$		1.2	mA
$AI_{ID}$	Idle mode: 87C552			50	$\mu A$
$AI_{PD}$	Power-down mode: 87C552	$2V < AV_{PD} < AV_{DD} \text{ max}$		50	$\mu A$

## Single-chip 8-bit microcontroller

87C552

## DC ELECTRICAL CHARACTERISTICS (Continued)

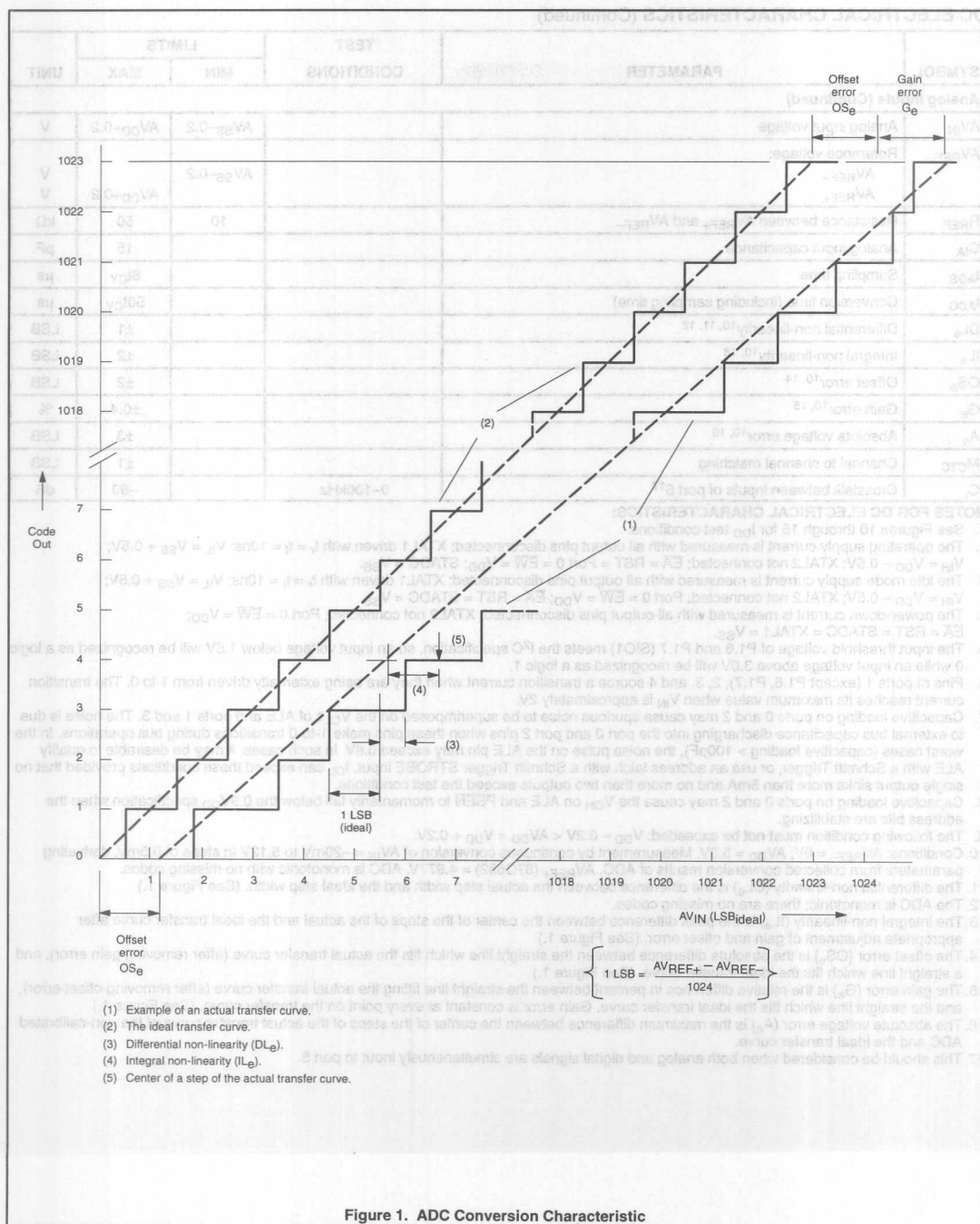
SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			MIN	MAX	
Analog Inputs (Continued)					
AV <sub>IN</sub>	Analog input voltage		AV <sub>SS</sub> -0.2	AV <sub>DD</sub> +0.2	V
AV <sub>REF</sub>	Reference voltage: AV <sub>REF-</sub> AV <sub>REF+</sub>		AV <sub>SS</sub> -0.2	AV <sub>DD</sub> +0.2	V V
R <sub>REF</sub>	Resistance between AV <sub>REF+</sub> and AV <sub>REF-</sub>		10	50	kΩ
C <sub>IA</sub>	Analog input capacitance			15	pF
t <sub>ADS</sub>	Sampling time			8t <sub>CY</sub>	μs
t <sub>ADC</sub>	Conversion time (including sampling time)			50t <sub>CY</sub>	μs
DL <sub>e</sub>	Differential non-linearity <sup>10, 11, 12</sup>			±1	LSB
IL <sub>e</sub>	Integral non-linearity <sup>10, 13</sup>			±2	LSB
OS <sub>e</sub>	Offset error <sup>10, 14</sup>			±2	LSB
G <sub>e</sub>	Gain error <sup>10, 15</sup>			±0.4	%
A <sub>e</sub>	Absolute voltage error <sup>10, 16</sup>			±3	LSB
M <sub>CTC</sub>	Channel to channel matching			±1	LSB
C <sub>t</sub>	Crosstalk between inputs of port 5 <sup>17</sup>	0-100kHz		-60	dB

## NOTES FOR DC ELECTRICAL CHARACTERISTICS:

- See Figures 10 through 15 for  $I_{DD}$  test conditions.
- The operating supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 10$ ns;  $V_{IL} = V_{SS} + 0.5$ V;  $V_{IH} = V_{DD} - 0.5$ V; XTAL2 not connected;  $EA = RST = Port\ 0 = EW = V_{DD}$ ;  $STADC = V_{SS}$ .
- The idle mode supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 10$ ns;  $V_{IL} = V_{SS} + 0.5$ V;  $V_{IH} = V_{DD} - 0.5$ V; XTAL2 not connected; Port 0 =  $EW = V_{DD}$ ;  $EA = RST = STADC = V_{SS}$ .
- The power-down current is measured with all output pins disconnected; XTAL2 not connected; Port 0 =  $EW = V_{DD}$ ;  $EA = RST = STADC = XTAL1 = V_{SS}$ .
- The input threshold voltage of P1.6 and P1.7 (SIO1) meets the I<sup>2</sup>C specification, so an input voltage below 1.5V will be recognized as a logic 0 while an input voltage above 3.0V will be recognized as a logic 1.
- Pins of ports 1 (except P1.6, P1.7), 2, 3, and 4 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when  $V_{IN}$  is approximately 2V.
- Capacitive loading on ports 0 and 2 may cause spurious noise to be superimposed on the  $V_{OLS}$  of ALE and ports 1 and 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100pF), the noise pulse on the ALE pin may exceed 0.8V. In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.  $I_{OL}$  can exceed these conditions provided that no single output sinks more than 5mA and no more than two outputs exceed the test conditions.
- Capacitive loading on ports 0 and 2 may cause the  $V_{OH}$  on ALE and PSEN to momentarily fall below the  $0.9V_{DD}$  specification when the address bits are stabilizing.
- The following condition must not be exceeded:  $V_{DD} - 0.2$ V <  $AV_{DD} < V_{DD} + 0.2$ V.
- Conditions:  $AV_{REF-} = 0$ V;  $AV_{DD} = 5.0$ V. Measurement by continuous conversion of  $AV_{IN} = -20$ mV to 5.12V in steps of 0.5mV, deriving parameters from collected conversion results of ADC.  $AV_{REF+}$  (87C552) = 4.977V. ADC is monotonic with no missing codes.
- The differential non-linearity ( $DL_e$ ) is the difference between the actual step width and the ideal step width. (See Figure 1.)
- The ADC is monotonic; there are no missing codes.
- The integral non-linearity ( $IL_e$ ) is the peak difference between the center of the steps of the actual and the ideal transfer curve after appropriate adjustment of gain and offset error. (See Figure 1.)
- The offset error ( $OS_e$ ) is the absolute difference between the straight line which fits the actual transfer curve (after removing gain error), and a straight line which fits the ideal transfer curve. (See Figure 1.)
- The gain error ( $G_e$ ) is the relative difference in percent between the straight line fitting the actual transfer curve (after removing offset error), and the straight line which fits the ideal transfer curve. Gain error is constant at every point on the transfer curve. (See Figure 1.)
- The absolute voltage error ( $A_e$ ) is the maximum difference between the center of the steps of the actual transfer curve of the non-calibrated ADC and the ideal transfer curve.
- This should be considered when both analog and digital signals are simultaneously input to port 5.

## Single-chip 8-bit microcontroller

## 87C552





## Single-chip 8-bit microcontroller

87C552

AC ELECTRICAL CHARACTERISTICS<sup>1, 2</sup>

SYMBOL	FIGURE	PARAMETER	12MHz CLOCK		16MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	
1/t <sub>CLCL</sub>	2	Oscillator frequency					3.5	16	MHz
t <sub>LHLL</sub>	2	ALE pulse width	127		85		2t <sub>CLCL</sub> -40		ns
t <sub>AVLL</sub>	2	Address valid to ALE low	28		8		t <sub>CLCL</sub> -55		ns
t <sub>LLAX</sub>	2	Address hold after ALE low	48		28		t <sub>CLCL</sub> -35		ns
t <sub>LLIV</sub>	2	ALE low to valid instruction in		234		150		4t <sub>CLCL</sub> -100	ns
t <sub>LLPL</sub>	2	ALE low to PSEN low	43		23		t <sub>CLCL</sub> -40		ns
t <sub>PLPH</sub>	2	PSEN pulse width	205		143		3t <sub>CLCL</sub> -45		ns
t <sub>PLIV</sub>	2	PSEN low to valid instruction in		145		83		3t <sub>CLCL</sub> -105	ns
t <sub>PXIX</sub>	2	Input instruction hold after PSEN	0		0		0		ns
t <sub>PXIZ</sub>	2	Input instruction float after PSEN		59		38		t <sub>CLCL</sub> -25	ns
t <sub>AVIV</sub>	2	Address to valid instruction in		312		208		5t <sub>CLCL</sub> -105	ns
t <sub>PLAZ</sub>	2	PSEN low to address float		10		10		10	ns
<b>Data Memory</b>									
t <sub>AVLL</sub>	3, 4	Address valid to ALE low	43		23		t <sub>CLCL</sub> -40		ns
t <sub>RLRH</sub>	3	RD pulse width	400		275		6t <sub>CLCL</sub> -100		ns
t <sub>WLWH</sub>	3	WR pulse width	400		275		6t <sub>CLCL</sub> -100		ns
t <sub>RLDV</sub>	3	RD low to valid data in		252		148		5t <sub>CLCL</sub> -165	ns
t <sub>RHDX</sub>	3	Data hold after RD	0		0		0		ns
t <sub>RHDZ</sub>	3	Data float after RD		97		55		2t <sub>CLCL</sub> -70	ns
t <sub>LLDV</sub>	3	ALE low to valid data in		517		350		8t <sub>CLCL</sub> -150	ns
t <sub>AVDV</sub>	3	Address to valid data in		585		398		9t <sub>CLCL</sub> -165	ns
t <sub>LLWL</sub>	3, 4	ALE low to RD or WR low	200	300	138	238	3t <sub>CLCL</sub> -50	3t <sub>CLCL</sub> +50	ns
t <sub>AVWL</sub>	3, 4	Address valid to WR low or RD low	203		120		4t <sub>CLCL</sub> -130		ns
t <sub>QVWX</sub>	4	Data valid to WR transition	23		3		t <sub>CLCL</sub> -60		ns
t <sub>DW</sub>	4	Data before WR	433		288		7t <sub>CLCL</sub> -150		ns
t <sub>WHQX</sub>	4	Data hold after WR	33		13		t <sub>CLCL</sub> -50		ns
t <sub>RLAZ</sub>	4	RD low to address float		0		0		0	ns
t <sub>WHLH</sub>	3, 4	RD or WR high to ALE high	43	123	23	103	t <sub>CLCL</sub> -40	t <sub>CLCL</sub> +40	ns
<b>External Clock</b>									
t <sub>CHCX</sub>	5	High time <sup>3</sup>	20		20		20		ns
t <sub>CLCX</sub>	5	Low time <sup>3</sup>	20		20		20		ns
t <sub>CLCH</sub>	5	Rise time <sup>3</sup>		20		20		20	ns
t <sub>CHCL</sub>	5	Fall time <sup>3</sup>		20		20		20	ns
<b>Serial Timing – Shift Register Mode<sup>4</sup> (Test Conditions: T<sub>amb</sub> = 0°C to +70°C; V<sub>SS</sub> = 0V; Load Capacitance = 80pF)</b>									
t <sub>XLXL</sub>	6	Serial port clock cycle time	1.0		0.75		12t <sub>CLCL</sub>		μs
t <sub>QVXH</sub>	6	Output data setup to clock rising edge	700		492		10t <sub>CLCL</sub> -133		ns
t <sub>XHDX</sub>	6	Output data hold after clock rising edge	50		8		2t <sub>CLCL</sub> -117		ns
t <sub>XHDX</sub>	6	Input data hold after clock rising edge	0		0		0		ns
t <sub>XHDV</sub>	6	Clock rising edge to input data valid		700		492		10t <sub>CLCL</sub> -133	ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- t<sub>CLCL</sub> = 1/f<sub>OSC</sub> = one oscillator clock period.  
t<sub>CLCL</sub> = 83.3ns at f<sub>OSC</sub> = 12MHz.  
t<sub>CLCL</sub> = 62.5ns at f<sub>OSC</sub> = 16MHz.
- These values are characterized but not 100% production tested.

## Single-chip 8-bit microcontroller

87C552

### AC ELECTRICAL CHARACTERISTICS (Continued)

SYMBOL	PARAMETER	INPUT	OUTPUT
<b>I<sup>2</sup>C Interface (Refer to Figure 9)<sup>5</sup></b>			
t <sub>HD</sub> ; STA	START condition hold time	≥ 14 t <sub>CLCL</sub>	> 4.0μs <sup>1</sup>
t <sub>LOW</sub>	SCL low time	≥ 16 t <sub>CLCL</sub>	> 4.7μs <sup>1</sup>
t <sub>HIGH</sub>	SCL high time	≥ 14 t <sub>CLCL</sub>	> 4.0μs <sup>1</sup>
t <sub>RC</sub>	SCL rise time	≤ 1μs	– <sup>2</sup>
t <sub>FC</sub>	SCL fall time	≤ 0.3μs	< 0.3μs <sup>3</sup>
t <sub>SU</sub> ; DAT1	Data set-up time	≥ 250ns	> 20 t <sub>CLCL</sub> – t <sub>RD</sub>
t <sub>SU</sub> ; DAT2	SDA set-up time (before rep. START cond.)	≥ 250ns	> 1μs <sup>1</sup>
t <sub>SU</sub> ; DAT3	SDA set-up time (before STOP cond.)	≥ 250ns	> 8 t <sub>CLCL</sub>
t <sub>HD</sub> ; DAT	Data hold time	≥ 0ns	> 8 t <sub>CLCL</sub> – t <sub>FC</sub>
t <sub>SU</sub> ; STA	Repeated START set-up time	≥ 14 t <sub>CLCL</sub>	> 4.7μs <sup>1</sup>
t <sub>SU</sub> ; STO	STOP condition set-up time	≥ 14 t <sub>CLCL</sub>	> 4.0μs <sup>1</sup>
t <sub>BUF</sub>	Bus free time	≥ 14 t <sub>CLCL</sub>	> 4.7μs <sup>1</sup>
t <sub>RD</sub>	SDA rise time	≤ 1μs	– <sup>2</sup>
t <sub>FD</sub>	SDA fall time	≤ 0.3μs	< 0.3μs <sup>3</sup>

NOTES:

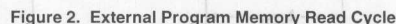
1. At 100 kbit/s. At other bit rates this value is inversely proportional to the bit-rate of 100 kbit/s.
2. Determined by the external bus-line capacitance and the external bus-line pull-resistor, this must be  $< 1\mu s$ .
3. Spikes on the SDA and SCL lines with a duration of less than  $3 \cdot t_{CLCL}$  will be filtered out. Maximum capacitance on bus-lines SDA and SCL = 400pF.
4.  $t_{CLCL} = 1/f_{OSC}$  = one oscillator clock period at pin XTAL1. For 62ns (42s)  $< t_{CLCL} < 285ns$  (16MHz (24Hz)  $> f_{OSC} > 3.5MHz$ ) the SiO1 interface meets the I<sup>2</sup>C-bus specification for bit-rates up to 100 kbit/s.
5. These values are guaranteed but not 100% production tested.

## 87C552

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

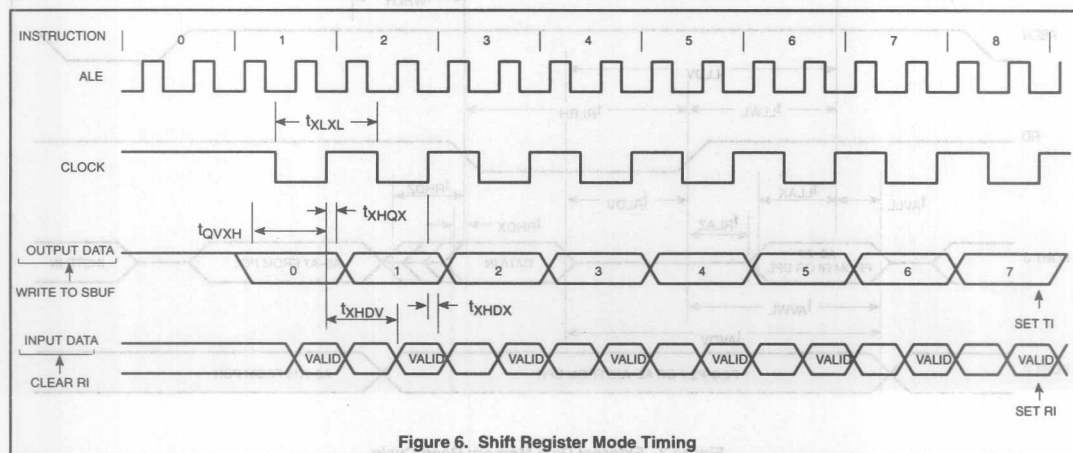
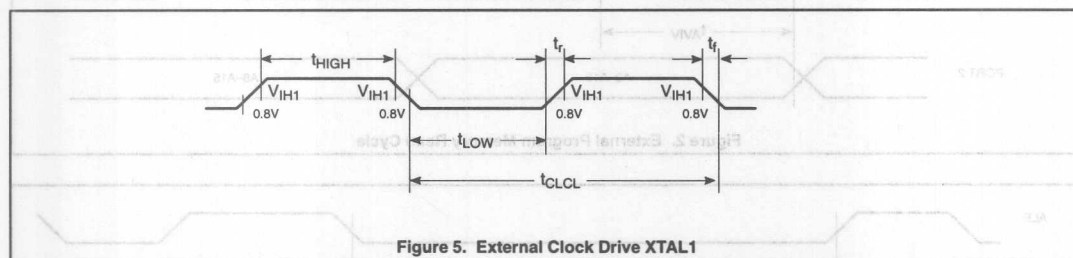
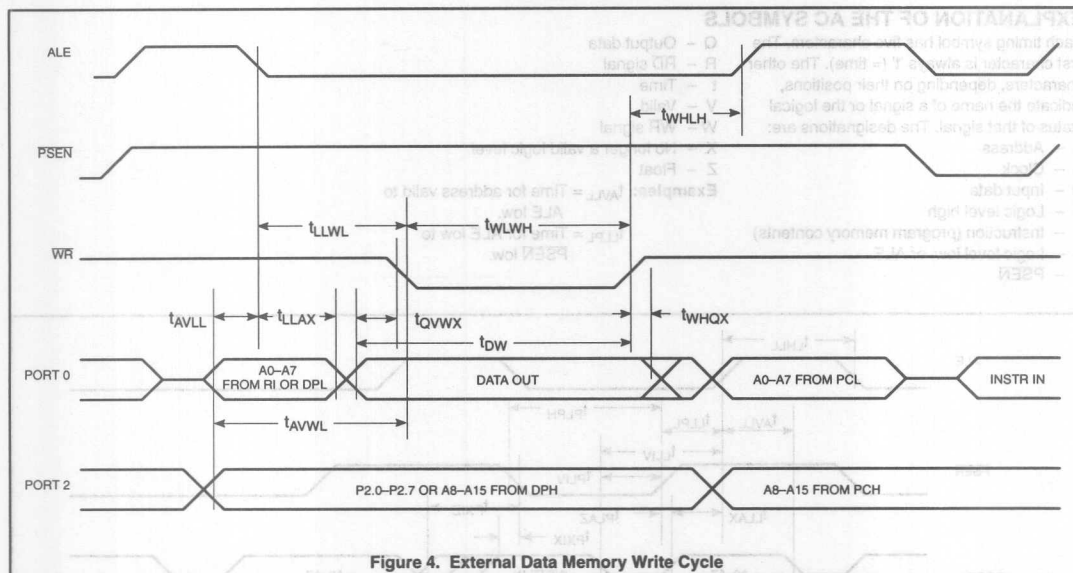
Q – Output data  
R –  $\overline{RD}$  signal  
t – Time  
V – Valid  
W –  $\overline{WR}$  signal  
X – No longer a valid logic level  
Z – Float

**Examples:**  $t_{AVLL}$  = Time for address valid to ALE low.  
 $t_{LLPL}$  = Time for ALE low to  $\overline{PSEN}$  low.



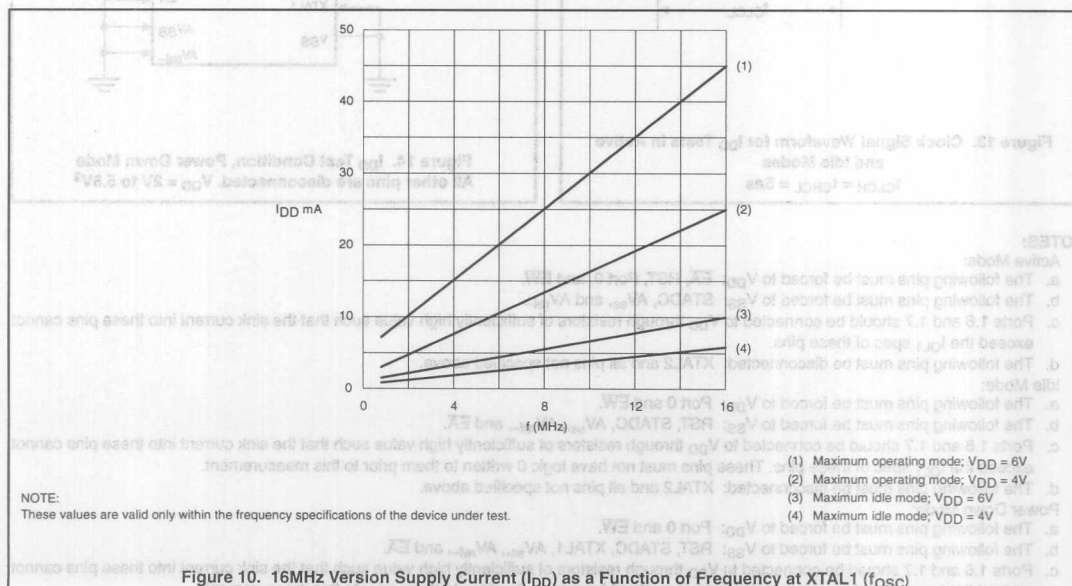
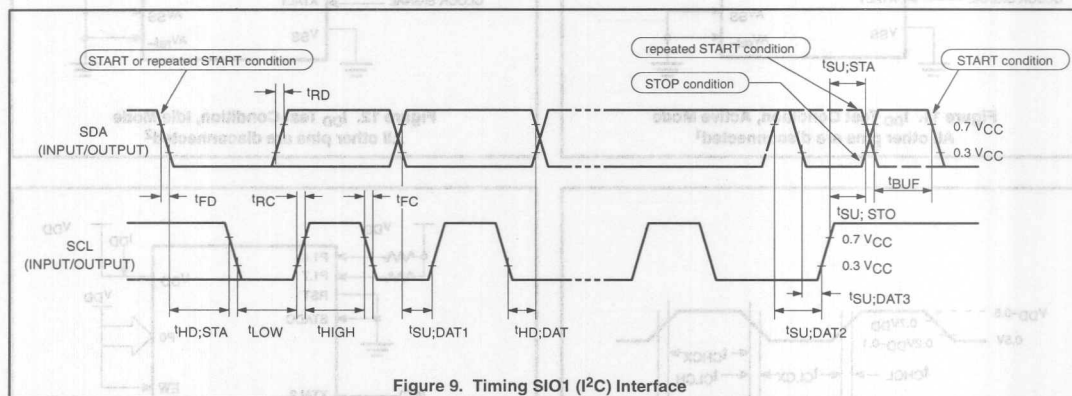
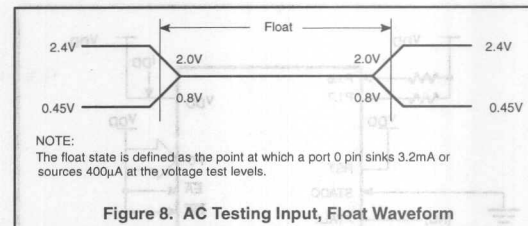
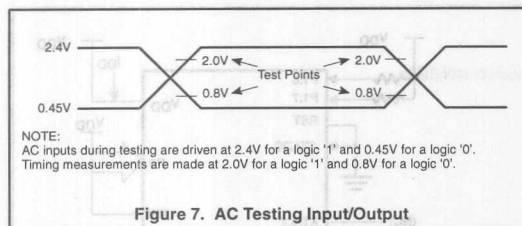
## Single-chip 8-bit microcontroller

87C552



## Single-chip 8-bit microcontroller

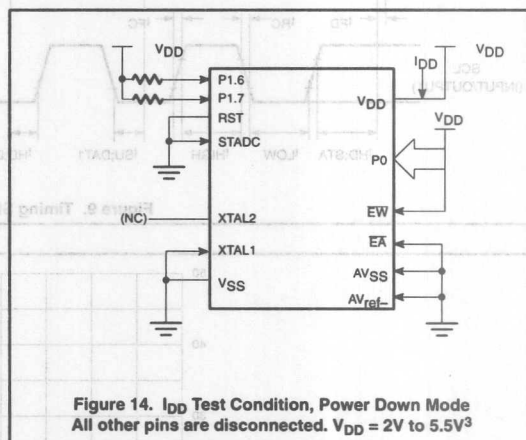
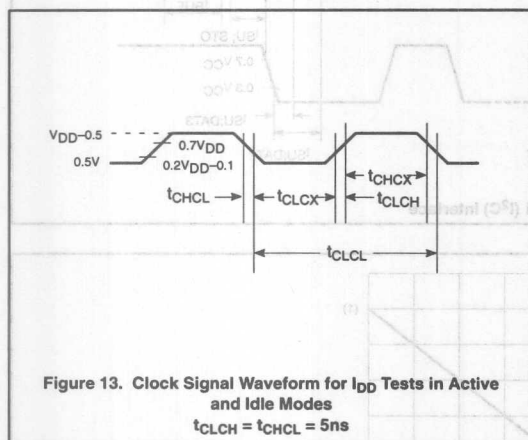
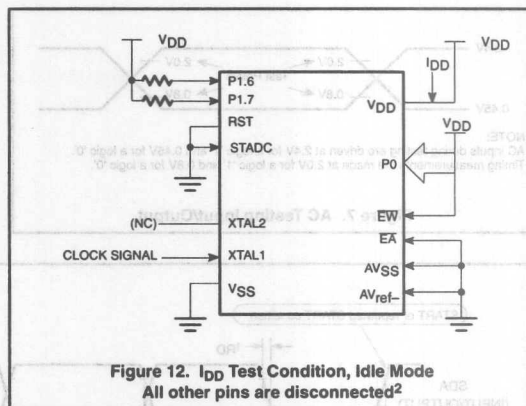
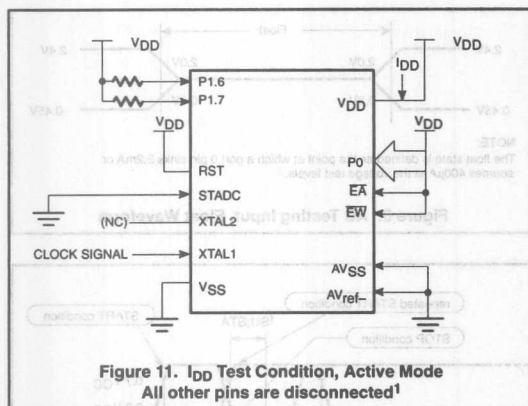
87C552





## Single-chip 8-bit microcontroller

## 87C552



## NOTES:

1. Active Mode:
  - a. The following pins must be forced to  $V_{DD}$ :  $\overline{EA}$ , RST, Port 0, and  $\overline{EW}$ .
  - b. The following pins must be forced to  $V_{SS}$ : STADC,  $AV_{SS}$ , and  $AV_{ref-}$ .
  - c. Ports 1.6 and 1.7 should be connected to  $V_{DD}$  through resistors of sufficiently high value such that the sink current into these pins cannot exceed the  $I_{OL1}$  spec of these pins.
  - d. The following pins must be disconnected: XTAL2 and all pins not specified above.
2. Idle Mode:
  - a. The following pins must be forced to  $V_{DD}$ : Port 0 and  $\overline{EW}$ .
  - b. The following pins must be forced to  $V_{SS}$ : RST, STADC,  $AV_{SS}$ ,  $AV_{ref-}$ , and  $\overline{EA}$ .
  - c. Ports 1.6 and 1.7 should be connected to  $V_{DD}$  through resistors of sufficiently high value such that the sink current into these pins cannot exceed the  $I_{OL1}$  spec of these pins. These pins must not have logic 0 written to them prior to this measurement.
  - d. The following pins must be disconnected: XTAL2 and all pins not specified above.
3. Power Down Mode:
  - a. The following pins must be forced to  $V_{DD}$ : Port 0 and  $\overline{EW}$ .
  - b. The following pins must be forced to  $V_{SS}$ : RST, STADC, XTAL1,  $AV_{SS}$ ,  $AV_{ref-}$ , and  $\overline{EA}$ .
  - c. Ports 1.6 and 1.7 should be connected to  $V_{DD}$  through resistors of sufficiently high value such that the sink current into these pins cannot exceed the  $I_{OL1}$  spec of these pins. These pins must not have logic 0 written to them prior to this measurement.
  - d. The following pins must be disconnected: XTAL2 and all pins not specified above.

## EPROM CHARACTERISTICS

The 87C552 is programmed by using a modified Quick-Pulse Programming™ algorithm. It differs from older methods in the value used for  $V_{PP}$  (programming supply voltage) and in the width and number of the ALE/PROG pulses.

The 87C552 contains two signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes identify the device as an 87C552 manufactured by Philips.

Table 3 shows the logic levels for reading the signature byte, and for programming the program memory, the encryption table, and the lock bits. The circuit configuration and waveforms for quick-pulse programming are shown in Figures 15 and 16. Figure 17 shows the circuit configuration for normal program memory verification.

### Quick-Pulse Programming

The setup for microcontroller quick-pulse programming is shown in Figure 15. Note that the 87C552 is running with a 4 to 6MHz oscillator. The reason the oscillator needs to be running is that the device is executing internal address and program data transfers.

The address of the EPROM location to be programmed is applied to ports 1 and 2, as shown in Figure 15. The code byte to be programmed into that location is applied to port 0. RST, PSEN, and pins of ports 2 and 3 specified in Table 3 are held at the "Program Code Data" levels indicated in Table 3. The ALE/PROG is pulsed low 25 times as shown in Figure 16.

To program the encryption table, repeat the 25-pulse programming sequence for

addresses 0 through 1FH, using the "Pgm Encryption Table" levels. Do not forget that after the encryption table is programmed, verification cycles will produce only encrypted data.

To program the lock bits, repeat the 25-pulse programming sequence using the "Pgm Lock Bit" levels. After one lock bit is programmed, further programming of the code memory and encryption table is disabled. However, the other lock bit can still be programmed.

Note that the  $EA/V_{PP}$  pin must not be allowed to go above the maximum specified  $V_{PP}$  level for any amount of time. Even a narrow glitch above that voltage can cause permanent damage to the device. The  $V_{PP}$  source should be well regulated and free of glitches and overshoot.

### Program Verification

If lock bit 2 has not been programmed, the on-chip program memory can be read out for program verification. The address of the program memory locations to be read is applied to ports 1 and 2 as shown in Figure 17. The other pins are held at the "Verify Code Data" levels indicated in Table 3. The contents of the address location will be emitted on port 0. External pull-ups are required on port 0 for this operation.

If the encryption table has been programmed, the data presented at port 0 will be the exclusive NOR of the program byte with one of the encryption bytes. The user will have to know the encryption table contents in order to correctly decode the verification data. The encryption table itself cannot be read out.

### Reading the Signature Bytes

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. The values are:

(030H) = 15H indicates manufactured by Philips Components

(031H) = 94H indicates 87C552

### Program/Verify Algorithms

Any algorithm in agreement with the conditions listed in Table 3, and which satisfies the timing specifications, is suitable.

### Erase Characteristics

Erase of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to the light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent erasure. **For this and secondary effects, it is recommended that an opaque label be placed over the window.** For elevated temperature or environments where solvents are being used, apply Kapton tape Fluorglas part number 2345-5, or equivalent.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 angstroms) to an integrated dose of at least 15W-sec/cm<sup>2</sup>. Exposing the EPROM to an ultraviolet lamp of 12,000μW/cm<sup>2</sup> rating for 20 to 39 minutes, at a distance of about 1 inch, should be sufficient. Erasure leaves the array in an all 1s state.

Table 3. EPROM Programming Modes

MODE	RST	PSEN	ALE/PROG	$EA/V_{PP}$	P2.7	P2.6	P3.7	P3.6
Read signature	1	0	1	1	0	0	0	0
Program code data	1	0	0*	$V_{PP}$	1	0	1	1
Verify code data	1	0	1	1	0	0	1	1
Pgm encryption table	1	0	0*	$V_{PP}$	1	0	1	0
Pgm lock bit 1	1	0	0*	$V_{PP}$	1	1	1	1
Pgm lock bit 2	1	0	0*	$V_{PP}$	1	1	0	0

#### NOTES:

1. 0 = Valid low for that pin; 1 = valid high for that pin.

2.  $V_{PP}$  = 12.75V ±0.25V.

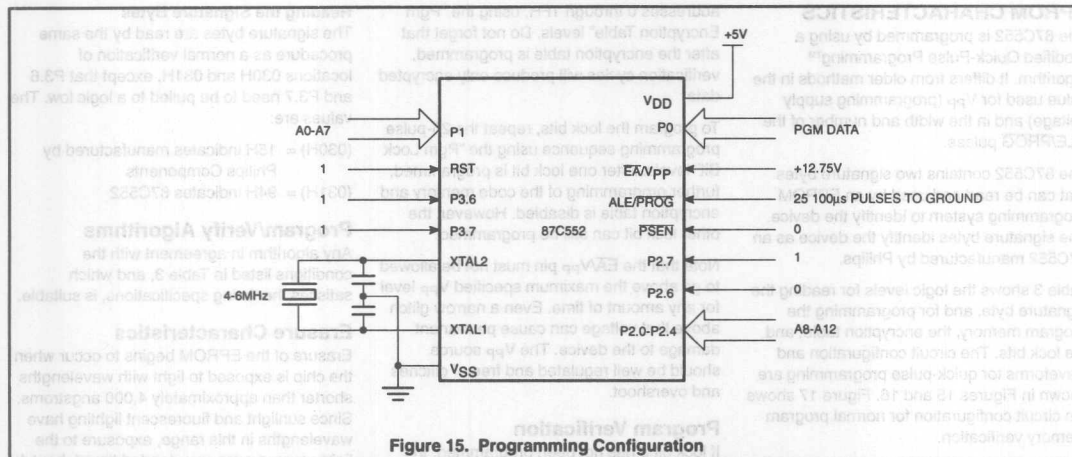
3.  $V_{DD}$  = 5V ±10% during programming and verification.

\* ALE/PROG receives 25 programming pulses while  $V_{PP}$  is held at 12.75V. Each programming pulse is low for 100μs (±10μs) and high for a minimum of 10μs.

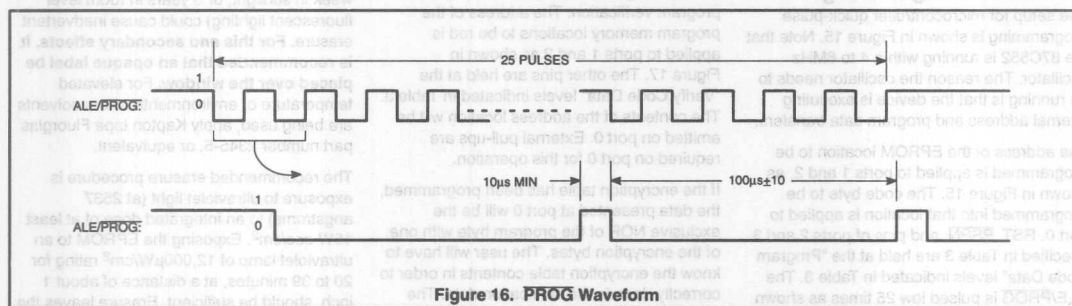
™Trademark phrase of Intel Corporation.

## Single-chip 8-bit microcontroller

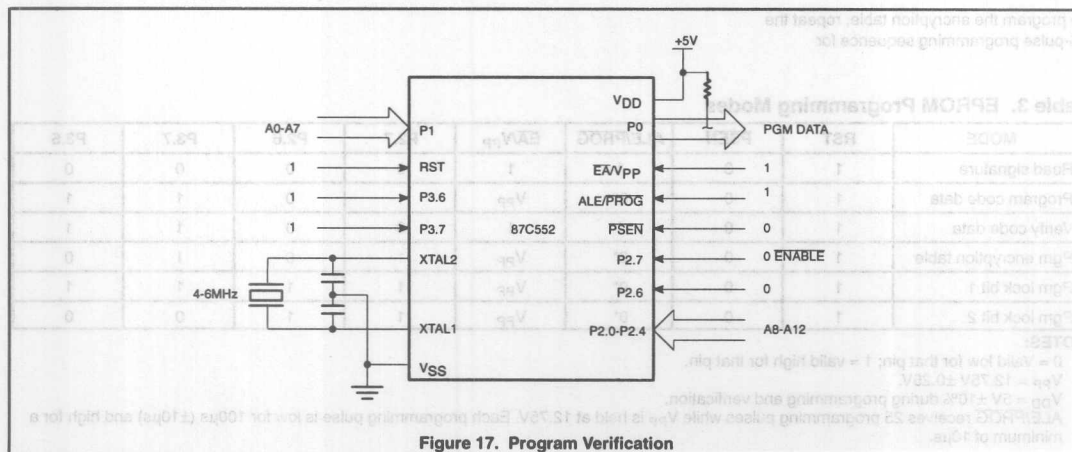
## 87C552



### Figure 15. Programming Configuration



**Figure 16. PROG Waveform**



**Figure 17. Program Verification**

## Single-chip 8-bit microcontroller

87C552

## EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

 $T_{amb} = 21^{\circ}\text{C}$  to  $+27^{\circ}\text{C}$ ,  $V_{DD} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$ 

SYMBOL	PARAMETER	MIN	MAX	UNIT
$V_{PP}$	Programming supply voltage	12.5	13.0	V
$I_{PP}$	Programming supply current		50	mA
$1/t_{CLCL}$	Oscillator frequency	4	6	MHz
$t_{AVGL}$	Address setup to PROG low	$48t_{CLCL}$		
$t_{GHAX}$	Address hold after PROG *	$48t_{CLCL}$		
$t_{DVGL}$	Data setup to PROG low	$48t_{CLCL}$		
$t_{GHDX}$	Data hold after PROG	$48t_{CLCL}$		
$t_{ESH}$	P2.7 (ENABLE) high to $V_{PP}$	$48t_{CLCL}$		
$t_{SHGL}$	$V_{PP}$ setup to PROG low	10		$\mu\text{s}$
$t_{GHSL}$	$V_{PP}$ hold after PROG	10		$\mu\text{s}$
$t_{GLGH}$	PROG width	90	110	$\mu\text{s}$
$t_{AVQV}$	Address to data valid	$48t_{CLCL}$		
$t_{ELQZ}$	ENABLE low to data valid	$48t_{CLCL}$		
$t_{EHQZ}$	Data float after ENABLE	0	$48t_{CLCL}$	
$t_{GHGL}$	PROG high to PROG low	10		$\mu\text{s}$

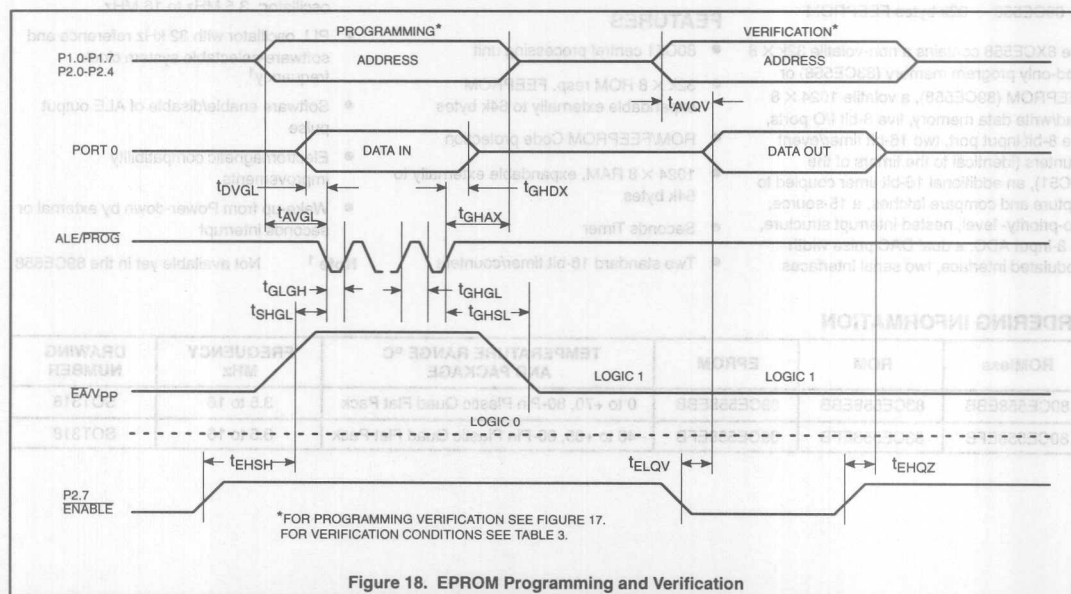
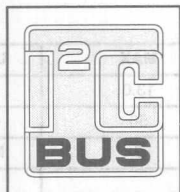


Figure 18. EPROM Programming and Verification



Purchase of Philips' I<sup>2</sup>C components conveys a license under the Philips' I<sup>2</sup>C patent to use the components in the I<sup>2</sup>C-system provided the system conforms to the I<sup>2</sup>C specifications defined by Philips.



## GENERAL DESCRIPTION

The 80CE558/83CE558/89CE558 (hereafter generically referred to as 8XCE558) Single-Chip 8-Bit Microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 8XCE558 has the same instruction set as the 80C51. Three versions of the derivative exist:

- 83CE558 — 32k bytes mask programmable ROM
- 80CE558 — ROMless version of the 83CE558
- 89CE558 — 32k bytes EEPROM

The 8XCE558 contains a non-volatile 32k × 8 read-only program memory (83CE558) or EEPROM (89CE558), a volatile 1024 × 8 read/write data memory, five 8-bit I/O ports, one 8-bit input port, two 16-bit timer/event counters (identical to the timers of the 80C51), an additional 16-bit timer coupled to capture and compare latches, a 15-source, two-priority-level, nested interrupt structure, an 8-input ADC, a dual DAC pulse width modulated interface, two serial interfaces

(UART and I<sup>2</sup>C-bus), a "watchdog" timer and on-chip oscillator and timing circuits. For systems that require extra capability, the 8XCE558 can be expanded using standard TTL compatible memories and logic.

In addition, the 8XCE558 has two software selectable modes of power reduction — idle mode and power-down mode. The idle mode freezes the CPU while allowing the RAM, timers, serial ports, and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.

The device also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of over 100 instructions: 49 one-byte, 45 two-byte, and 17 three-byte. With a 16 MHz system clock, 58% of the instructions are executed in 0.75μs and 40% in 1.5μs. Multiply and divide instructions require 3μs.

## FEATURES

- 80C51 central processing unit
- 32k × 8 ROM resp. EEPROM expandable externally to 64k bytes
- ROM/EEPROM Code protection
- 1024 × 8 RAM, expandable externally to 64k bytes
- Seconds Timer
- Two standard 16-bit timer/counters
- An additional 16-bit timer/counter coupled to four capture registers and three compare registers
- A 10-bit ADC with eight multiplexed analog inputs and programmable autoscan
- Two 8-bit resolution, pulse width modulation outputs
- Five 8-bit I/O ports plus one 8-bit input port shared with analog inputs
- I<sup>2</sup>C-bus serial I/O port with byte oriented master and slave functions
- Full-duplex UART compatible with the standard 80C51
- On-chip watchdog timer
- 15 interrupt sources with 2 priority levels (2 to 6 external sources possible)
- Extended temperature range (−40 to +85°C)
- 4.5 to 5.5V supply voltage range
- Frequency range for 80C51 standard oscillator: 3.5 MHz to 16 MHz
- PLL oscillator with 32 kHz reference and software-selectable system clock frequency<sup>1</sup>
- Software enable/disable of ALE output pulse
- Electromagnetic compatibility improvements
- Wake-up from Power-down by external or seconds interrupt

**Note 1** Not available yet in the 89CE558

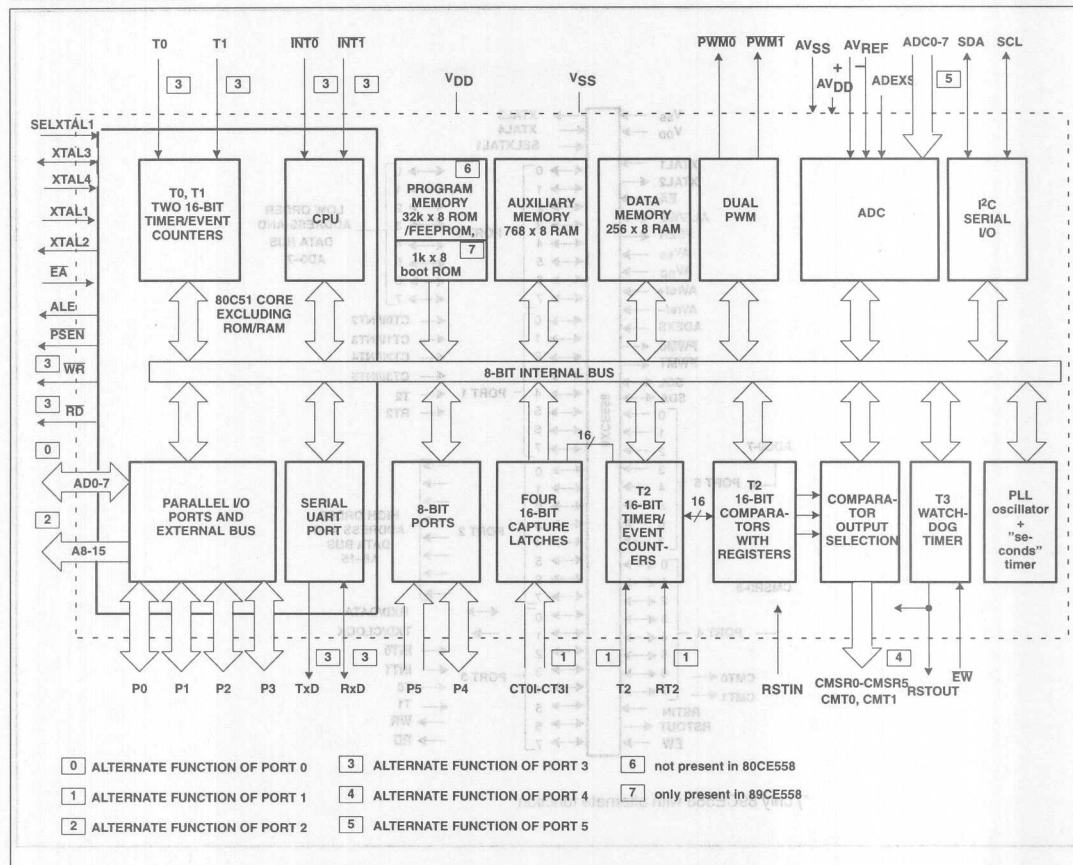
## ORDERING INFORMATION

ROMless	ROM	EPROM	TEMPERATURE RANGE °C AND PACKAGE	FREQUENCY MHz	DRAWING NUMBER
80CE558EBB	83CE558EBB	89CE558EBB	0 to +70, 80-Pin Plastic Quad Flat Pack	3.5 to 16	SOT318
80CE558EFB	83CE558EFB	89CE558EFB	−40 to +85, 80-Pin Plastic Quad Flat Pack	3.5 to 16	SOT318



# BLOCK DIAGRAM

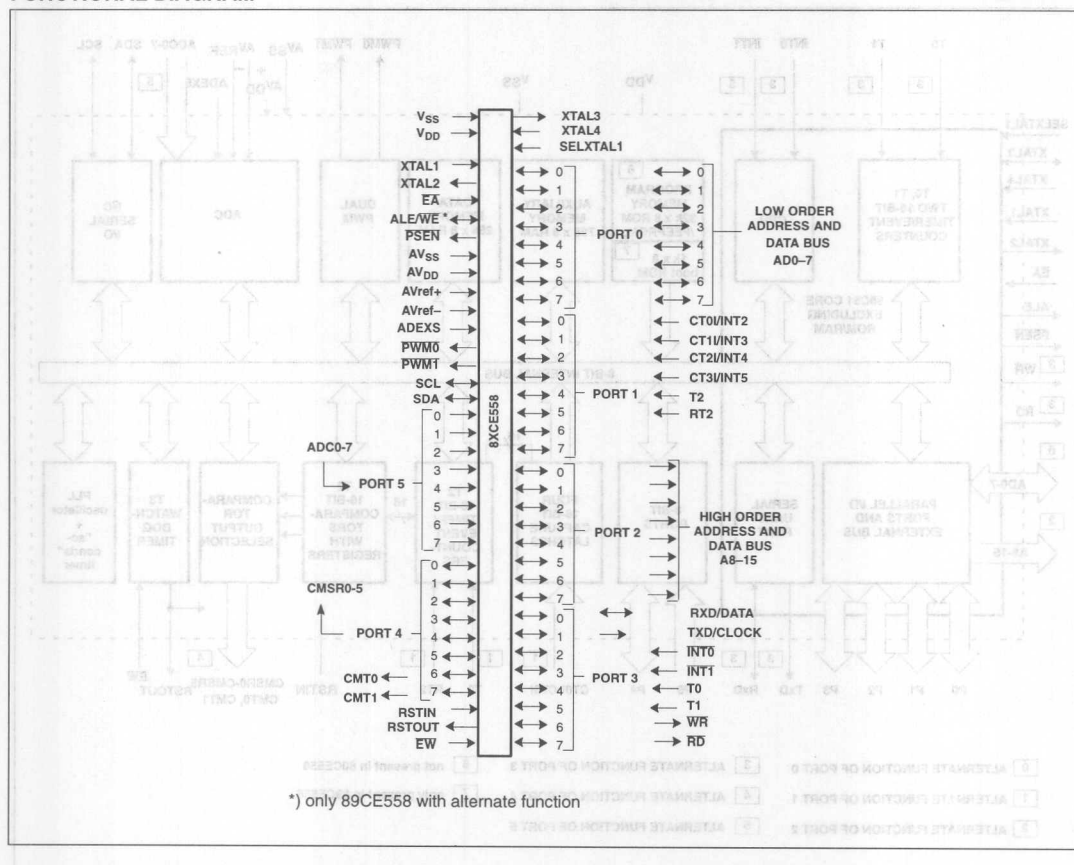
FUNCTIONAL BLOCK DIAGRAM



## Single-chip 8-bit microcontroller

80CE558/83CE558/89CE558

## FUNCTIONAL DIAGRAM



## Single-chip 8-bit microcontroller

80CE558/83CE558/89CE558

## PINNING DIAGRAM

Pin	Function	Pin	Function
1	AVref-	42	P3.1/TXD/CLOCK
2	AVref+	43	P3.2/INT0
3	AVss1	44	P3.3/INT1
4	AVDD1	45	P3.4/T0
5	P5.7/ADC7	46	P3.5/T1
6	P5.6/ADC6	47	P3.6/WR
7	P5.5/ADC5	48	P3.7/RD
8	P5.4/ADC4	49	NC
9	P5.3/ADC3	50	NC
10	P5.2/ADC2	51	XTAL2
11	P5.1/ADC1	52	XTAL1
12	P5.0/ADC0	53	VDD3
13	VSS1	54	VSS3
14	VDD1	55	P2.0/A8
15	ADEXS	56	P2.1/A9
16	PWM0	57	P2.2/A10
17	PWM1	58	P2.3/A11
18	EW	59	P2.4/A12
19	P4.0/CMSR0	60	P2.5/A13
20	P4.1/CMSR1	61	P2.6/A14
21	P4.2/CMSR2	62	P2.7/A15
22	P4.3/CMSR3	63	PSEN
23	RSTOUT	64	ALE/WE*
24	P4.4/CMSR4	65	E <sub>A</sub>
25	P4.5/CMSR5	66	VDD4
26	P4.6/CMT0	67	VSS4
27	P4.7/CMT1	68	P0.7/AD7
28	VDD2	69	P0.6/AD6
29	VSS2	70	P0.5/AD5
30	RSTIN	71	P0.4/AD4
31	P1.0/CT0/INT2	72	P0.3/AD3
32	P1.1/CT1/INT3	73	P0.2/AD2
33	P1.2/CT2/INT4	74	P0.1/AD1
34	P1.3/CT3/INT5	75	P0.0/AD0
35	P1.4/T2	76	AVDD2
36	P1.5/RT2	77	AVSS2
37	P1.6	78	XTAL3
38	P1.7	79	XTAL4
39	SCL	80	SELXTAL1
40	SDA		
41	P3.0/RXD/DATA		

\* only 89CE558 with alternative function

# PIN DESCRIPTION

SYMBOL	PIN	DESCRIPTION
AV <sub>ref-</sub>	1	Low end of analog to digital conversion reference resistor
AV <sub>ref+</sub>	2	High end of analog to digital conversion reference resistor.
AV <sub>SS1</sub>	3	Analog ground for ADC
AV <sub>DD1</sub>	4	Analog power supply (+5V) for ADC
AV <sub>SS2</sub>	77	Analog ground; for PLL oscillator
AV <sub>DD2</sub>	76	Analog power supply; (+ 5V) for PLL oscillator
P5.7–P5.0	5–12	<b>Port 5</b> 8-bit input port Port pin Alternative function P5.0–P5.7 Eight input channels to ADC (ADC0–ADC7)
V <sub>DD1</sub> –4		<b>Digital power supply:</b> +5V power supply pins during normal operation and power reduction modes. All pins must be connected.
V <sub>SS1</sub> –4		<b>Digital ground:</b> circuit ground potential. All pins must be connected.
V <sub>DD1</sub>	14	<b>Digital power supply</b>
V <sub>SS1</sub>	13	
V <sub>DD2</sub>	28	<b>Digital power supply</b>
V <sub>SS2</sub>	29	
V <sub>DD3</sub>	53	<b>Digital power supply</b>
V <sub>SS3</sub>	54	
V <sub>DD4</sub>	66	<b>Digital power supply</b>
V <sub>SS4</sub>	67	
ADEXS	15	<b>Start ADC operation:</b> Input starting analog to digital conversion triggered by a programmable edge (ADC operation can also be started by software). This pin must not float.
PWM0	16	<b>Pulse width modulation</b> output 0
PWM1	17	<b>Pulse width modulation</b> output 1
EW	18	<b>Enable watchdog timer:</b> Enable for T3 watchdog timer and disable Power-down mode. This pin must not float.
P4.0 – P4.7	19 – 22 24 – 27	<b>Port 4</b> 8-bit quasi-bidirectional I/O port Port pin Alternative function P4.0 CMSR0 } P4.1 CMSR1 } P4.2 CMSR2 } Timer T2: compare and set/reset P4.3 CMSR3 } outputs on a match with timer T2 P4.4 CMSR4 } P4.5 CMSR5 } P4.6 CMT0 } Timer T2: compare and toggle outputs P4.7 CMT1 } on a match with timer T2
RSTIN	30	<b>Reset:</b> Input to reset the 8xC5558.
RSTOUT	23	<b>Reset:</b> Output of the 8xC558 for resetting peripheral devices during initialization and Watchdog Timer overflow.
P1.0 – P1.7	31 – 38	<b>Port 1</b> 8-bit quasi-bidirectional I/O port Port pin Alternative function P1.0 CT0/INT2 } P1.1 CT1/INT3 } Capture timer inputs for P1.2 CT2/INT4 } timer T2 or external interrupt inputs P1.3 CT3/INT5 } P1.4 T2 : T2 event input, rising edge triggered P1.5 RT2 : T2 timer reset input, rising edge triggered P1.6 P1.7
SCL	39	<b>I<sup>2</sup>C-bus serial clock</b> I/O port

## 8 Single-chip 8-bit microcontroller

## 80CE558/83CE558/89CE558

## PIN DESCRIPTION (Continued)

SYMBOL	PIN	DESCRIPTION
SDA	40	<b>I<sup>2</sup>C-bus serial data I/O port</b> If SCL and SDA are not connected to I <sup>2</sup> C-buslines, their input levels should be defined otherwise (e.g. by a pull-up resistor to V <sub>DD</sub> ) in order to prevent these port input stages from floating, which could affect the total power consumption. Especially during Power-down Mode extra power consumption by floating inputs is an adverse effect.
P3.0 – P3.7	41 – 48	<b>8-bit quasi-bidirectional I/O port</b> <b>Port pin Alternative function</b> P3.0 RXD : Serial input port P3.1 TXD : Serial output port P3.2 INT0 : External interrupt P3.3 INT1 : External interrupt P3.4 T0 : Timer 0 external input P3.5 T1 : Timer 1 external input P3.6 WR : External data memory write strobe P3.7 RD : External data memory read strobe
XTAL2	51	<b>Crystal pin 2:</b> output of the inverting amplifier that forms the oscillator. Left open-circuit when an external oscillator clock is used.
XTAL1	52	<b>Crystal pin 1:</b> input to the inverting amplifier that forms the oscillator, and input to the internal clock generator. Receives the external oscillator clock signal when an external oscillator is used. Must be connected to logic HIGH if the PLL oscillator is selected (SELXTAL1 = LOW)
P2.0 – P2.7	55 – 62	<b>Port2:</b> 8-bit quasi-bidirectional I/O port with internal pull-ups. During access to external memories (RAM/ROM) that use 16-bit addresses (MOVX@DPTR) Port 2 emits the high order address byte. The alternative function of P2.7 for the 89CE558 is the output enable signal for verify/read modes (active low). Port 2 can sink/source one TTL (=4 LSTTL) input. It can drive CMOS inputs without external pull-ups.
PSEN	63	<b>Program Store Enable output:</b> read strobe to the external program memory via Port 0 and 2. Is activated twice each machine cycle during fetches from external program memory. When executing out of external program memory two activations of PSEN are skipped during each access to external data memory. PSEN is not activated (remains HIGH) during no fetches from external program memory. PSEN can sink/source 8 LSTTL inputs. It can drive CMOS inputs without external pull-ups.
ALE/WE	64	<b>Address Latch Enable output:</b> latches the low byte of the address during access of external memory in normal operation. It is activated every six oscillator periods except during an external data memory access. ALE/WE can sink/-source 8 LSTTL inputs. It can drive CMOS inputs without an external pull-up. The alternative function for the 89CE558 is the programming pulse input WE.  To prohibit the toggling of ALE pin (RFI noise reduction) the bit RFI in the PCON Register (PCON.5) must be set by software. This bit is cleared on RESET and can be set and cleared by software. When set, ALE pin will be pulled down internally, switching an external address latch to a quiet state. The MOVX instruction will still toggle ALE as a normal MOVX. ALE will retain its normal high value during Idle mode and a low value during Power-down mode while in the "RFI" mode. Additionally during internal access (EA = 1) ALE will toggle normally when the address exceeds the internal program memory size. During external access (EA = 0) ALE will always toggle normally, whether the flag "RFI" is set or not.
EA	65	<b>External Access Input:</b> When, during RESET, EA is held at a TTL HIGH level the CPU executes out of the internal program memory, provided the program counter is less than 32768. When EA is held at a TTL low level during RESET, the CPU executes out of external program memory via Port 0 and Port 2. EA is not allowed to float. EA is latched during RESET and don't care after RESET.
P0.7–P0.0	68 – 75	<b>Port 0:</b> 8-bit open drain bidirectional I/O port. It is also the multiplexed low-order address and data bus during accesses to external memory (during these accesses internal pull-ups are activated). Port 0 can sink/source 8 LSTTL inputs.
XTAL3	78	<b>Crystal pin,</b> output of the inverting amplifier that forms the 32 KHz oscillator
XTAL4	79	<b>Crystal pin,</b> input to the inverting amplifier that forms the 32 KHz oscillator, must be connected to logic LOW if the PLL oscillator is not selected (SELXTAL1 = HIGH).
SELXTAL1	80	<b>Must be connected to logic HIGH level to select the HF oscillator, using the XTAL1/XTAL2 crystal. If pulled low the PLL is selected for clocking of the controller, using the XTAL3/XTAL4 crystal.</b>

To avoid a 'latch-up' effect at Power-on, the voltage on any pin at any time must not be higher or lower than V<sub>DD</sub>+ 0.5V or V<sub>SS</sub>- 0.5V respectively.



## 8 Single-chip 8-bit microcontroller

## 80CE558/83CE558/89CE558

### ELECTROMAGNETIC COMPATIBILITY (EMC) IMPROVEMENTS

Primary attention was paid on the reduction of electromagnetic emission of the microcontroller 8xCE558.

The following features effect in reducing the electromagnetic emission and additionally improve the electromagnetic susceptibility:

- Four supply voltage pins ( $V_{DD}$ ) and four ground pins ( $V_{SS}$ ) with a pair of  $V_{DD}$  and  $V_{SS}$  at two adjacent pins in the center at one side of the package and at two adjacent pins at the opposite side of the package and one more  $V_{SS}$  pin at each of the other two sides of the package
- Separated  $V_{DD}$  pins for the internal logic and the port buffers
- Internal decoupling capacitance improves the EMC radiation behavior and the EMC immunity
- External capacitors are to be located as close as possible between pins  $V_{DD1}$  and  $V_{SS1}$ ,  $V_{DD2}$  and  $V_{SS2}$ ,  $V_{DD3}$  and  $V_{SS3}$  as well as  $V_{DD4}$  and  $V_{SS4}$ ; ceramic chip capacitors are recommended (100nF).

Useful in applications that require no external memory or temporarily no external memory:

- The ALE output signal (pulses at a frequency of  $f_{CLK}/6$ ) can be disabled under software control (bit 5 in the SFR PCON: "RFI"); if disabled, no ALE pulse will occur. ALE pin will be pulled down internally, switching an external address latch to a quiet state. The MOVX instruction will still toggle ALE (external data memory is accessed). ALE will retain its normal HIGH value during Idle mode and a LOW value during Power-down mode while in the "RFI" reduction mode. Additionally during internal access ( $EA = 1$ ) ALE will toggle normally when the address exceeds the internal program memory size. During external access ( $EA = 0$ ) ALE will always toggle normally, whether the flag "RFI" is set or not.

### 18.0 FUNCTIONAL DESCRIPTION

#### General

The 8xCE558 is a stand-alone high-performance microcontroller designed for use in real time applications such as instrumentation, industrial control, medium to high-end consumer applications and specific automotive control applications.

In addition to the 80C51 standard functions, the device provides a number of dedicated hardware functions for these applications.

The 8xCE558 is a control-oriented CPU with on-chip program and data memory. It can be extended with external program memory up to 64 K bytes. It can also access up to 64 K bytes of external data memory. For systems requiring extra capability, the 8xCE558 can be expanded using standard memories and peripherals.

The 8xCE558 has two software selectable modes of reduced activity for further power reduction – Idle and Power-down. The Idle mode freezes the CPU while allowing the RAM, timers, serial ports and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator causing all other chip functions to be inoperative. The Power-down mode can be terminated by an external Reset and by any one of the two external interrupts. (see description Wake-up from Power-down mode).

#### 1.1 Memory organization

The central processing unit (CPU) manipulates operands in three memory spaces; these are the 64 K-byte external data memory, 1024 byte internal data memory (consisting of 256 bytes standard RAM and 768 bytes AUX-RAM) and the 64 K-byte internal and external program memory (see Figure 1).

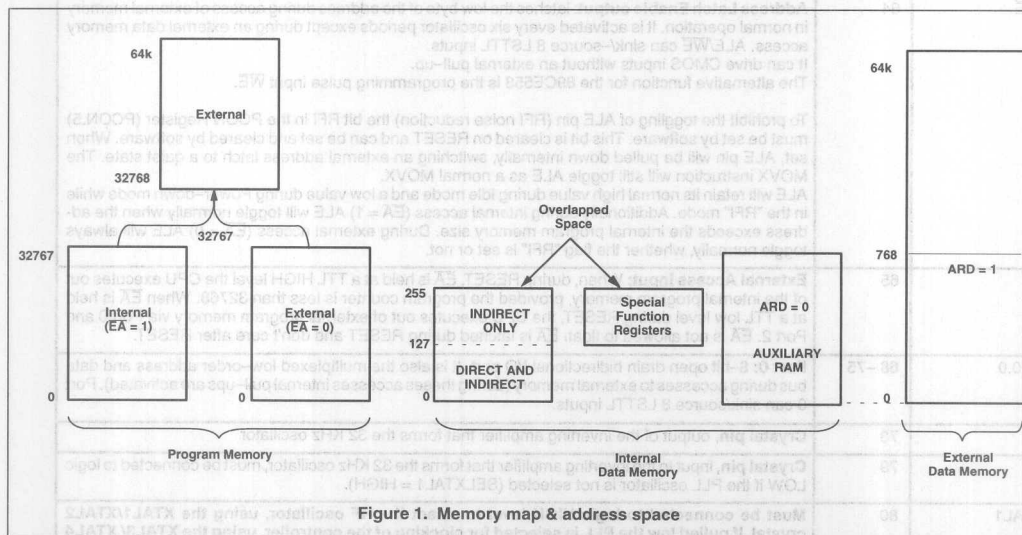


Figure 1. Memory map & address space

## Single-chip 8-bit microcontroller

## 80CE558/83CE558/89CE558

## 1.1.1 Program Memory

The program memory of the 8xCE558 consists of 32 Kbyte ROM resp. EEPROM ("Flash Memory") on-chip, externally expandable up to 64 Kbyte. If the EA pin is held HIGH, the 8xCE558 executes out of the internal program memory unless the address exceeds 7FFFH. Locations 8000H through 0FFFFH are then fetched from the external program memory. If the EA pin was set LOW during RESET the 8xCE558 fetches all instructions from the external program memory. The EA input is latched during RESET and is don't care after RESET.

By setting a mask programmable security bit (ROM) resp. by software programmable security byte (EEPROM) the internal program memory content is protected i.e. it cannot be read out at any time by any test mode or by any instruction in the external program memory space. The MOVX instructions are the only which have access to program code in the internal or external program memory. The EA input is latched during RESET and is 'don't care' after RESET. This implementation prevents from reading internal program code by switching from external program memory to internal program memory during MOVX instruction or an instruction that handles immediate data.

Table 1 lists the access to the internal and external program memory with MOVX instructions when the security feature has been activated.

## 1.1.2 Internal Data Memory

The internal data memory is divided into three physically separated parts:

256 byte of RAM, 768 byte of AUX-RAM, and a 128-byte special function area. These can be addressed each in a different way (see also Table 2).

- RAM 0 to 127 can be addressed directly and indirectly as in the 80C51. Address pointers are R0 and R1 of the selected registerbank.
- RAM 128 to 255 can only be addressed indirectly. Address pointers are R0 and R1 of the selected registerbank.
- AUX-RAM 0 to 767 is indirectly addressable as external DATA MEMORY locations 0 to 767 via MOVX-Datapointer instruction, unless it is disabled by setting ARD = 1.
- AUX-RAM 0 to 767 is indirectly addressable via pageregister (XRAMP) and MOVX-R1 instructions, unless it is disabled by setting ARD = 1 (see Figure 2). When executing from internal program memory, an access to

AUX-RAM 0 to 767 will not affect the ports P0, P2, P3.6 and P3.7. An access to external DATA MEMORY locations higher than 767 will be performed with the MOVX @ DPTR instructions in the same way as in the 80C51 structure, so with P0 and P2 as data/address bus and P3.6 and P3.7 as write and read timing signals. Note that the external DATA MEMORY cannot be accessed with R0 and R1 as address pointer if the AUX-RAM is enabled (ARD = 0, default).

The Special Function Registers (SFR) can only be addressed directly in the address range from 128 to 255 (see table 4).

- Four 8-register banks occupy locations 0 through 31 in the lower RAM area. Only one of these banks may be enabled at a time. The next 16 bytes, locations 32 through 47, contain 128 directly addressable bit locations. The stack can be located anywhere in the internal 256 byte RAM. The stack depth is only limited by the available internal RAM space of 256 bytes (see Figure 3). All registers except the program counter and the four 8-register banks reside in the Special Function Register address space.

Table 1.

	Access to Internal Program Memory	Access to External Program Memory
MOVC in internal program memory	YES	YES
MOVC in external program memory	NO	YES

If the security feature has not been activated, there are no restrictions for MOVX instructions.

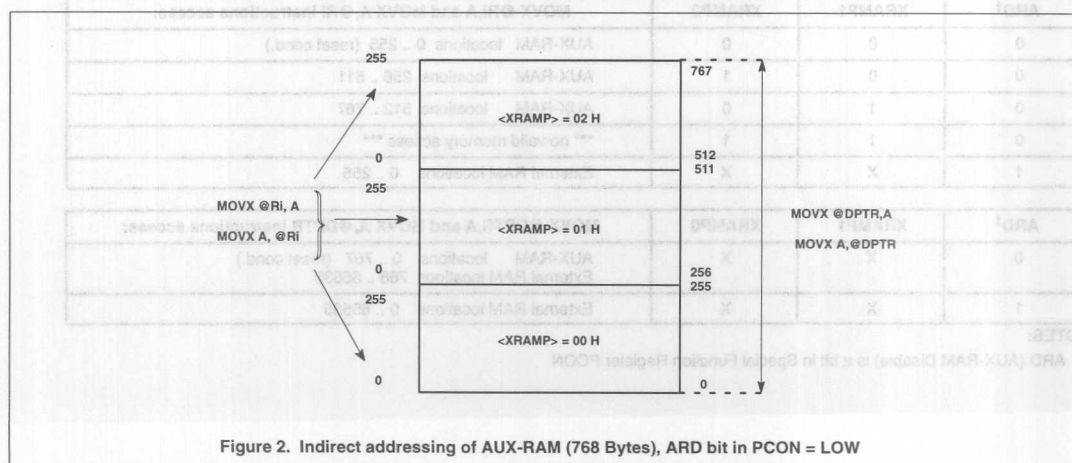


Figure 2. Indirect addressing of AUX-RAM (768 Bytes), ARD bit in PCON = LOW

## Single-chip 8-bit microcontroller

## 80CE558/83CE558/89CE558

Table 2. AUX-RAM 0 to 255

LOCATION	ADDRESSED
RAM 0 to 127	Direct and Indirect
AUX-RAM 0 to 767	Indirect Only with MOVX
RAM 128 to 255	Indirect Only
SFR 128 to 255	Direct Only

## AUX-RAM Page Register XRAMP

The AUX-RAM Page Register is used to select one of three 256-byte pages of the internal 768-byte AUX-RAM for MOVX-accesses via R0 or R1. Its reset value is (XXXXXX00).

	7	6	5	4	3	2	1	0
XRAMP (0FAH)	x	x	x	x	x	x	XRAMP1	XRAMP0

x: undefined during read, a write operation must write "0" to these location

BIT	SYMBOL	FUNCTION
XRAMP.7	-	(reserved for future use <sup>1</sup> )
XRAMP.6	-	(reserved for future use <sup>1</sup> )
XRAMP.5	-	(reserved for future use <sup>1</sup> )
XRAMP.4	-	(reserved for future use <sup>1</sup> )
XRAMP.3	-	(reserved for future use <sup>1</sup> )
XRAMP.2	-	(reserved for future use <sup>1</sup> )
XRAMP.1	XRAMP1	AUX-RAM page select bit 1
XRAMP.0	XRAMP0	AUX-RAM page select bit 0

## NOTES:

- User software should not write 1s to reserved bits. These bits may be used in future 80C51 family products to invoke new features. In that case, the reset or inactive value of the new bit will be LOW, and its active value will be HIGH. The value read from a reserved bit is indeterminate.

Table 3 shows the memory locations for all possible MOVX-accesses:

Table 3.

ARD <sup>1</sup>	XRAMP1	XRAMP0	MOVX @Ri,A and MOVX A,@Ri instructions access:
0	0	0	AUX-RAM locations 0 .. 255 (reset cond.)
0	0	1	AUX-RAM locations 256 .. 511
0	1	0	AUX-RAM locations 512 .. 767
0	1	1	*** no valid memory access ***
1	X	X	External RAM locations 0 .. 255

ARD <sup>1</sup>	XRAMP1	XRAMP0	MOVX @DPTR,A and MOVX A,@DPTR instructions access:
0	X	X	AUX-RAM locations 0 .. 767 (reset cond.) External RAM locations 768 .. 65535
1	X	X	External RAM locations 0 .. 65535

## NOTES:

- ARD (AUX-RAM Disable) is a bit in Special Function Register PCON

## 8 Single-chip 8-bit microcontroller

## 80CE558/83CE558/89CE558

Table 4. Special Function Register Memory Map and Reset Values

HIGH NIBBLE OF SFR ADDRESS								
LOW	8	9	A	B	C	D	E	F
0	P0 % 11111111	P1 % 11111111	P2 % 11111111	P3 % 11111111	P4 % 11111111	PSW % 00000000	ACC % 00000000	B % 00000000
1	SP 00001111							
2	DPL 00000000							
3	DPH 00000000							
4								
5								
6	ADRSLO # XXXXXXXX	ADRS1 # XXXXXXXX	ADRS2 # XXXXXXXX	ADRS3 # XXXXXXXX	ADRS4 # XXXXXXXX	ADRS5 # XXXXXXXX	ADRS6 # XXXXXXXX	ADRS7 # XXXXXXXX
7	PCON 00000000				P5 # XXXXXXXX	ADCON 00000000	ADPSS 00000000	ADRS8 # 000000XX
8	TCON % 00000000	S0CON % 00000000	IEN0 % 00000000	IP0 % X0000000	TM2IR % 00000000	S1CON % 00000000	IEN1 % 00000000	IP1 % 00000000
9	TMOD 00000000	S0BUF XXXXXXXX	CML0 00000000		CMH0 00000000	S1STA # 11111000		PLLCON 00001101
A	TL0 00000000		CML1 00000000		CMH1 00000000	S1DAT 00000000	TM2CON 00000000	XRAMP XXXXXXXX00
B	TL1 00000000		CML2 00000000		CMH2 00000000	S1ADR 00000000	CTCON 00000000	FMCON * 000X0000
C	TH0 00000000		CTL0 # XXXXXXXX		CTH0 # XXXXXXXX		TML2 # 00000000	PWM0 00000000
D	TH1 00000000		CTL1 # XXXXXXXX		CTH1 # XXXXXXXX		TMH2 # 00000000	PWM1 00000000
E			CTL2 # XXXXXXXX		CTH2 # XXXXXXXX		STE 11000000	PWMP 00000000
F			CTL3 # XXXXXXXX		CTH3 # XXXXXXXX		RTE 00000000	T3 00000000

% = Bit addressable register

# = Read only register

X = Undefined

\* = only in 89CE558

## 1.2 Addressing

The 8xCE558 has five modes for addressing:

- Register
- Direct
- Register-Indirect
- Immediate
- Base-Register plus  
Index-Register-Indirect

The first three methods can be used for addressing destination operands. Most

instructions have a "destination/source" field that specifies the data type, addressing methods and operands involved. For operations other than MOVs, the destination operand is also a source operand.

Access to memory addresses is as follows:

- Register in one of the four 8-register banks through Register, Direct or Register-Indirect addressing
- 1024 bytes of internal RAM through Direct or Register-Indirect addressing. Bytes 0-127 of internal RAM may be addressed directly/indirectly. Bytes

128-255 of internal RAM share their address location with the SFRs and so may only be addressed indirectly as data RAM. Bytes 0-767 of AUX-RAM can only be addressed indirectly via MOVX.

- Special Function Register through direct addressing at address locations 128-255 (see Figure 4).
- External data memory through Register-Indirect addressing
- Program memory look-up tables through Base-Register plus Index-Register-Indirect addressing

## Single-chip 8-bit microcontroller

## 80CE558/83CE558/89CE558

Byte Address (Hex)	Bit Address (Hex)	Byte Address (Decimal)	Direct Byte Address (Hex)	Bit Address (Hex)	Register Mnemonic
FFH	(MSB)	255	FFH	(MSB)	
				PT2 PCM2 PCM1 PCM0 PCT3 PCT2 PCT1 PCT0	IP1
			F8H	FF FE FD FC FB FA F9 F8	
2FH	7F 7E 7D 7C 7B 7A 79 78	47	F0H	F7 F6 F5 F4 F3 F2 F1 F0	B
2EH	77 76 75 74 73 72 71 70	46		ET2 ECM2 ECM ECM0 ECT3 ECT2 ECT1 ECT0	
2DH	6F 6E 6D 6C 6B 6A 69 68	45	E8H	EF EE ED EC EB EA E9 E8	IEN1
2CH	67 66 65 64 63 62 61 60	44			
2BH	5F 5E 5D 5C 5B 5A 59 58	43	E0H	E7 E6 E5 E4 E3 E2 E1 E0	ACC
2AH	57 56 55 54 53 52 51 50	42		CR2 ENS1 STA STO SI AA CR1 CR0	
29H	4F 4E 4D 4C 4B 4A 49 48	41	D8H	DF DE DD DC DB DA D9 D8	S1CON
28H	47 46 45 44 43 42 41 40	40		CY AC F0 RS1 RS0 OV F1 P	
27H	3F 3E 3D 3C 3B 3A 39 38	39	D0H	D7 D6 D5 D4 D3 D2 D1 D0	PSW
26H	37 36 35 34 33 32 31 30	38		T2OV CMI2 CMI1 CMI0 CTI3 CTI2 CTI1 CTI0	
25H	2F 2E 2D 2C 2B 2A 29 28	37	C8H	CF CE CD CC CB CA C9 C8	TM2IR
24H	27 26 25 24 23 22 21 20	36			
23H	1F 1E 1D 1C 1B 1A 19 18	35	C0H	C7 C6 C5 C4 C3 C2 C1 C0	P4
22H	17 16 15 14 13 12 11 10	34		PAD PS1 PS0 PT1 PX1 PTO PX0	
21H	0F 0E 0D 0C 0B 0A 09 08	33	B8H	BF BE BD BC BB BA B9 B8	IP0
20H	07 06 05 04 03 02 01 00	32			
1FH		31	B0H	B7 B6 B5 B4 B3 B2 B1 B0	P3
	Bank 3			EA EAD ES1 ES0 ET1 EX1 ET0 EX0	
18H		24	A8H	AF AE AD AC AB AA A9 A8	IEN0
17H	Bank 2	23			
			A0H	A7 A6 A5 A4 A3 A2 A1 A0	P2
10H		16		SM0 SM1 SM2 REN TB8 RB8 TI RI	
0FH	Bank 1	15	98H	9F 9E 9D 9C 9B 9A 99 98	S0CON
08H	Bank 0	8	90H	97 96 95 94 93 92 91 90	P1
				TF1 TR1 TF0 TR0 IE1 IT1 IE0 IT0	
07H		7	88H	8F 8E 8D 8C 8B 8A 89 88	TCON
00H		0	80H	87 86 85 84 83 82 81 80	P0

Figure 3. RAM Bit Addresses

Figure 4. Special Function Register Bit Addresses

## 1.3 I/O facilities

The 8xCE558 has six 8-bit ports. Ports 0 to 3 are the same as in the 80C51, with the exception of the additional functions of Port 1. The parallel I/O function of Port 4 is equal to that of Ports 1, 2 and 3. Port 5 has a parallel input port function, but has no function as an output port.

The SDA and SCL lines serve the serial port SIO1 (I<sup>2</sup>C). Because the I<sup>2</sup>C-bus may be active while the device is disconnected from V<sub>DD</sub>, these pins, are provided with open drain drivers.

Ports 0, 1, 2, 3, 4 and 5 perform the following alternative functions:

- Port 0: provides the multiplexed low-order address and data bus used for expanding the 8xCE558 with standard memories and peripherals;
- Port 1: Port 1 is used for a number of special functions:
  - 4 capture inputs (or external interrupt request inputs if capture information is not utilized)
  - external counter input
  - external counter reset input

- Port 2: provides the high-order address bus when the 8xCE558 is expanded with external Program Memory and/or external Data Memory.
- Port 3: pins can be configured individually to provide:
  - external interrupt request inputs
  - counter inputs
  - receiver input and transmitter output of serial port SIO 0 (UART)
  - control signals to read and write external Data Memory



## 8 Single-chip 8-bit microcontroller

## 80CE558/83CE558/89CE558

- Port 4: can be configured to provide signals indicating a match between timer counter T2 and its compare registers.
- Port 5: may be used in conjunction with the ADC interface. Unused analog inputs can be used as digital inputs. As Port 5 lines may be used as inputs to the ADC, these digital inputs have an inherent hysteresis to prevent the input logic from drawing too much current from the power lines when driven by analog signals. Channel to channel crosstalk should be taken into consideration when both digital and analog signals are simultaneously input to Port 5 (see DC characteristics).

All ports are bidirectional with the exception of Port 5 which is an input port.

Pins of which the alternative function is not used may be used as normal bidirectional I/Os.

The generation or use of a Port 1, Port 3 or Port 4 pin as an alternative function is carried out automatically by the 8xCE558 provided the associated Special Function Register bit is set HIGH.

The pull-up arrangements of Ports 1 - 5 are shown in Figure 5.

#### 1.4 Pulse Width Modulated Outputs

The 8xCE558 contains two pulse width modulated output channels (see Figure 6). These channels generate pulses of programmable length and interval. The

repetition frequency is defined by an 8-bit prescaler PWMP, which supplies the clock for the counter. The prescaler and counter are common to both PWM channels. The 8-bit counter counts modulo 255, i.e., from 0 to 254 inclusive. The value of the 8-bit counter is compared to the contents of two registers: PWM0 and PWM1. Provided the contents of either of these registers is greater than the counter value, the corresponding PWM0 or PWM1 output is set LOW. If the contents of these registers are equal to, or less than the counter value, the output will be HIGH. The pulse-width-ratio is therefore defined by the contents of the registers PWM0 and PWM1. The pulse-width-ratio is in the range of 0/255 to 255/255 and may be programmed in increments of 1/255.

Buffered PWM outputs may be used to drive DC motors. The rotation speed of the motor would be proportional to the contents of PWMn. The PWM outputs may also be configured as a dual DAC. In this application, the PWM outputs must be integrated using conventional operational amplifier circuitry. If the resulting output voltages have to be accurate, external buffers with their own analog supply should be used to buffer the PWM outputs before they are integrated. The repetition frequency fpwm, at the PWMn outputs is given by:

$$fpwm = \frac{f_{CLK}}{2 \times (1 + PWMP) \times 255}$$

This gives a repetition frequency range of 123Hz to 31.4kHz ( $f_{CLK} = 16\text{MHz}$ ). By

loading the PWM registers with either 00H or FFH, the PWM channels will output a constant HIGH or LOW level, respectively. Since the 8-bit counter counts modulo 255, it can never actually reach the value of the PWM registers when they are loaded with FFH.

When a compare register (PWM0 or PWM1) is loaded with a new value, the associated output is updated immediately. It does not have to wait until the end of the current counter period. Both PWMn output pins are driven by push-pull drivers. These pins are not used for any other purpose.

Prescaler frequency control register PWMP

PWMP (FEH)

7	6	5	4	3	2	1	0
MSB				LSB			

Bit Function

PWMP.0-7 Prescaler division factor = (PWMP) + 1.

PWM0 (FCH)

PWM1 (FDH)

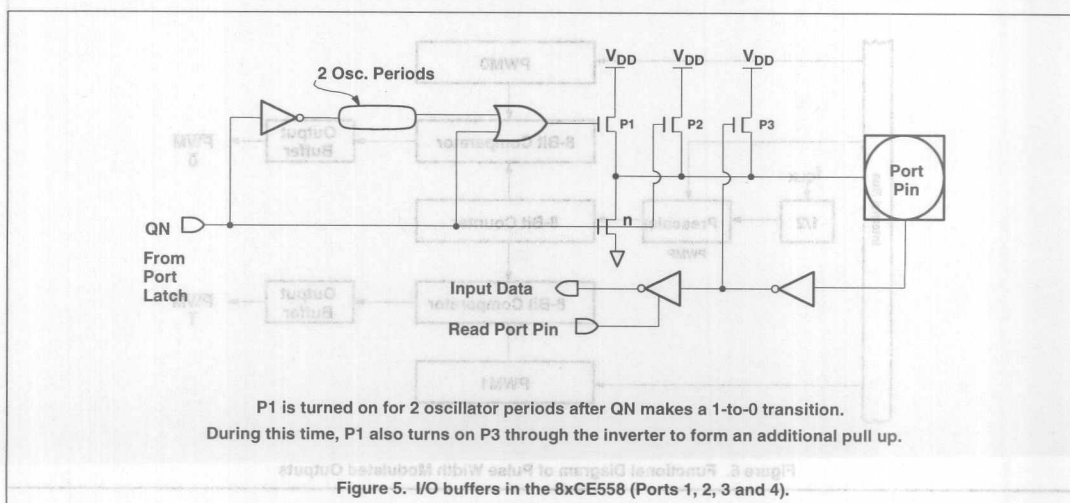
7	6	5	4	3	2	1	0
MSB				LSB			

Reading PWMP gives the current reload value. The actual count of the prescaler cannot be read.

Bit Function

PWM0.0-7 Low/high ratio of

PWM1.0-7  $PWMn = \frac{(PWMn)}{255 - (PWMn)}$



## Single-chip 8-bit microcontroller

## 80CE558/83CE558/89CE558

## 1.5 Analog / Digital Converter

In order to have a minimum of ADC service overhead in the microcontroller program, the ADC is able to operate autonomously by its configurable autoscan feature and a set of 8 buffer registers (10 bit), which store the result of the conversion of each analog input channel. The functional diagram of the ADC is shown in Figure 7.

## Feature Overview:

- 10-bit resolution.
- 8 multiplexed analog inputs.
- Programmable autoscan of the analog inputs.
- Bit oriented 8-bit scan-select register to select analog inputs.
- Continuous scan or one time scan configurable from 1 to 8 analog inputs.
- Start of a conversion by software or with an external signal.
- Programmable prescaler (div by 2, 4, 6, 8) to adapt to different oscillator frequencies.
- Eight 10-bit buffer registers, one register for each analog input channel.
- Conversion time for one A/D conversion: 15µs ... 50µs
- Differential non-linearity : DLe 1 LSB.
- Integral non-linearity : ILLe 2 LSB.
- Offset error : OSLe 2 LSB.
- Gain error : Ge 0.4 %.

- Absolute voltage error : Ae 3 LSB.
- Channel to channel matching : Mctc 1 LSB.
- Crosstalk between analog inputs : Ct < -60dB.  
@ 10.0 kHz.
- Monotonic and no missing codes.
- Separated analog (AVDD, AVSS) and digital (VDD, VSS) supply voltages.
- Reference voltage at two special pins : AVREF- and AVREF+.

## 1.5.1 Functional description:

Table 5.

ADCON	A/D control register	read/write
ADPSS	Analog port scan-select register	read/write
ADRSLn	8 A/D result registers, contain the 8 lower bits	read only
ADRSnH	A/D result register, contains the 2 higher bits	read only
P5	Digital input port (shared with analog inputs)	read only

After a RESET of the microcontroller the ADCON and ADPSS register bits are initialized to zero, registers ADRSLn and ADRSH are undefined.

ADRSLn and ADRSH are specified as read only registers to prevent conflict situations between software write operation and writing the A/D conversion result in ADRSn by the hardware.

## 1.5.2 Idle and Power-down Mode for the A/D converter

If the Idle or Power-down Mode is activated, then an A/D conversion in progress is aborted, the ADSST flag is cleared and the internal clock is halted. The interrupt flag ADINT will not be set. The ADRSn registers 10 bit buffer are not affected.

Table 6 shows resultant conversion times (tconv) at external clock frequencies (fCLK) and ADC prescaler divisors (m), which are programmable by the bits ADCON.7 and ADCON.6. For conversion times outside the specified range the specified characteristics are not guaranteed; those conversion times are put in brackets.

Table 6. Configuration time configurable examples (tconv/ms)

fCLK m	6MHz	8MHz	12MHz	16MHz
2	26	19.5	[13]	[9.75]
4	50	37.5	25	18.75
6	[74]	[55.5]	37	27.75
8	[98]	[73.5]	49	36.75

conversion time = 6 x m + 1 machine cycles

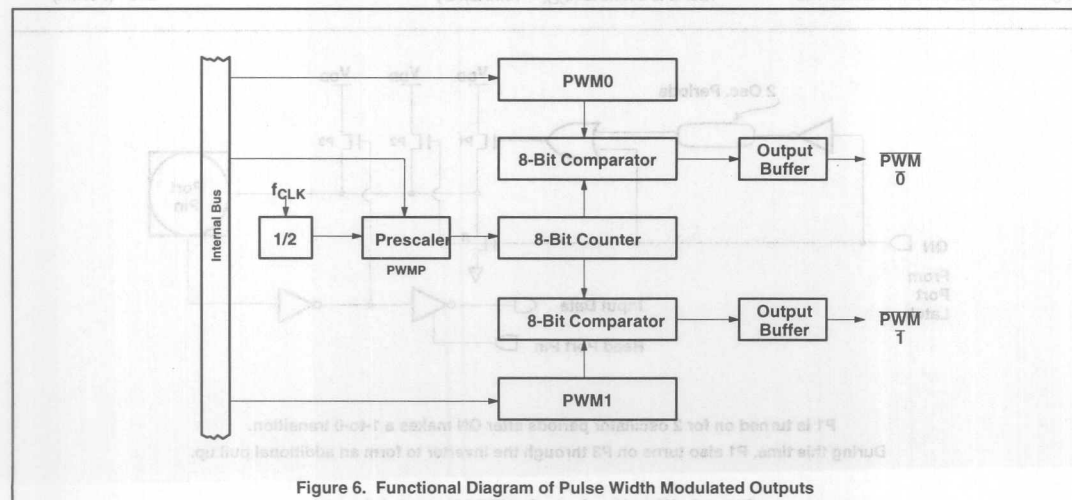
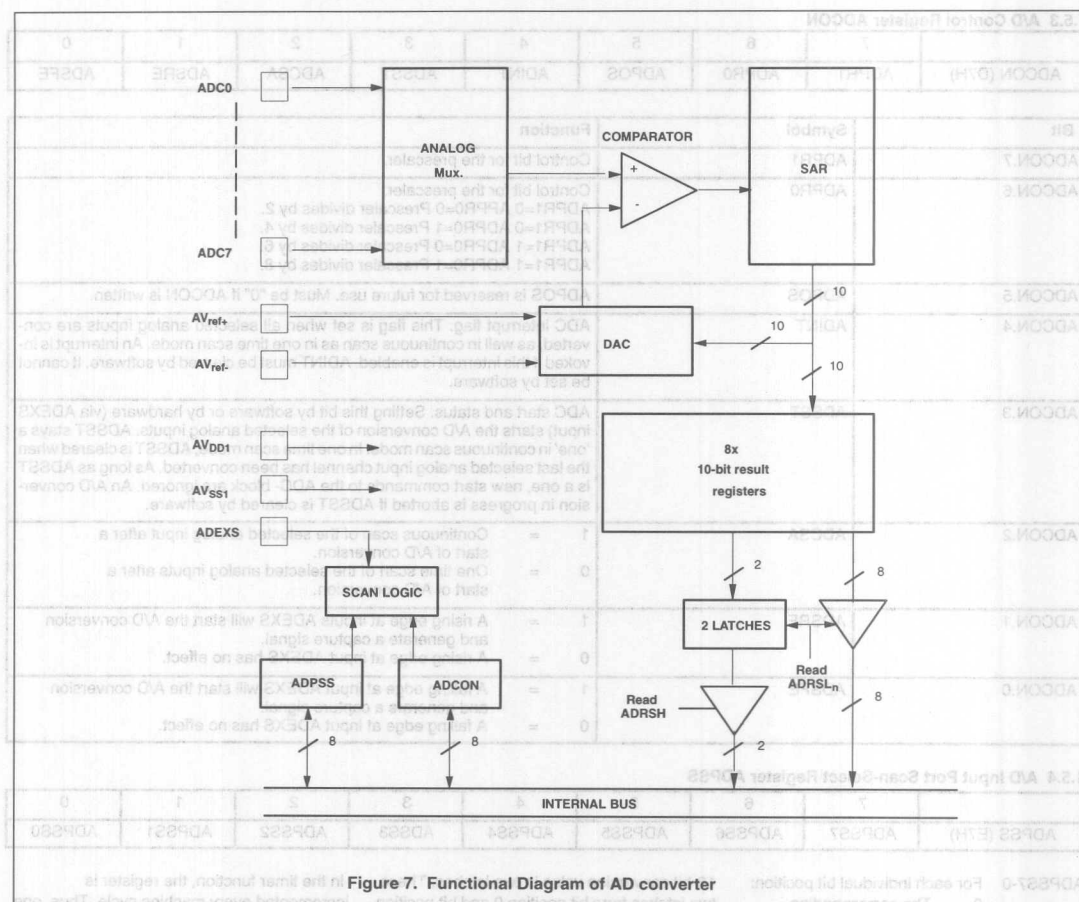


Figure 6. Functional Diagram of Pulse Width Modulated Outputs

## Single-chip 8-bit microcontroller

80CE558/83CE558/89CE558



**Figure 7. Functional Diagram of AD converter**

oscillator frequency.  
oscillator periods, the count rate is 1/12 of the  
Since a machine cycle consists of 12  
can think of it as counting machine cycles.

ADRSH are read only registers. ADRSL and ADRSH are all 8 bit read registers. The upper 8 bits of

in the auto-learn loop.  
analysis input is included  
The corresponding  
the auto-learn loop.

## Single-chip 8-bit microcontroller

80CE558/83CE558/89CE558

## 1.5.3 A/D Control Register ADCON

3.5.5 A/D Control Register ADCON								
	7	6	5	4	3	2	1	0
ADCON (D7H)	ADPR1	ADPR0	ADPOS	ADINT	ADSST	ADCSA	ADSRE	ADSFE
Bit	Symbol	Function						
ADCON.7	ADPR1	Control bit for the prescaler.						
ADCON.6	ADPR0	Control bit for the prescaler. ADPR1=0 ADPR0=0 Prescaler divides by 2. ADPR1=0 ADPR0=1 Prescaler divides by 4. ADPR1=1 ADPR0=0 Prescaler divides by 6. ADPR1=1 ADPR0=1 Prescaler divides by 8.						
ADCON.5	ADPOS	ADPOS is reserved for future use. Must be "0" if ADCON is written.						
ADCON.4	ADINT	ADC interrupt flag. This flag is set when all selected analog inputs are converted, as well in continuous scan as in one time scan mode. An interrupt is invoked if this interrupt is enabled. ADINT must be cleared by software. It cannot be set by software.						
ADCON.3	ADSST	ADC start and status. Setting this bit by software or by hardware (via ADEXS input) starts the A/D conversion of the selected analog inputs. ADSST stays a "one" in continuous scan mode. In one time scan mode, ADSST is cleared when the last selected analog input channel has been converted. As long as ADSST is a one, new start commands to the ADC- block are ignored. An A/D conversion in progress is aborted if ADSST is cleared by software.						
ADCON.2	ADCSA	1 = Continuous scan of the selected analog input after a start of A/D conversion. 0 = One time scan of the selected analog inputs after a start of A/D conversion.						
ADCON.1	ADSRE	1 = A rising edge at inputs ADEXS will start the A/D conversion and generate a capture signal. 0 = A rising edge at input ADEXS has no effect.						
ADCON.0	ADSFE	1 = A falling edge at input ADEXS will start the A/D conversion and generate a capture signal. 0 = A falling edge at input ADEXS has no effect.						

## 1.5.4 A/D Input Port Scan-Select Register ADPSS

	7	6	5	4	3	2	1	0
ADPSS (E7H)	ADPSS7	ADPSS6	ADPSS5	ADPSS4	ADSS3	ADPSS2	ADPSS1	ADPSS0

ADPSS7-0 For each individual bit position:  
 0 = The corresponding analog input is skipped in the auto-scan loop.  
 1 = The corresponding analog input is included in the auto-scan loop.

If all bits are 'zero' then no A/D conversion can be started. If ADPSS is written while an A/D conversion is in progress (ADSST in the ADCON register is 'one') then the auto-scan loop with the previous selected analog inputs is completed first. The next auto-scan loop is performed with the new selected analog inputs.

## 1.5.5 A/D Result Registers ADRLn and ADRSH:

There are 8 ADRLn registers and one ADRSH register. Reading an ADRLn register by software copies at the same time the two highest bits of the corresponding

10-bit conversion value in two latches. These two latches form bit position 0 and bit position 1 of register ADRSH. The upper 6 bits of ADRSH are all 0's if read. ADRLn and ADRSH are read only registers.

## 1.6 Timer / Counters

The 8xCE558 contains three 16-bit timer/event counters: Timer 0, Timer 1 and Timer T2 and one 8-bit timer, T3. Timer 0 and Timer 1 may be programmed to carry out the following functions:

- Measure time intervals and pulse durations
- Count events
- Generate interrupt requests

## 1.6.1 Timer 0 and Timer 1

Timers 0 and 1 each have a control bit in SFR TMOD that selects the timer or counter function of the corresponding timer.

In the timer function, the register is incremented every machine cycle. Thus, one can think of it as counting machine cycles. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In the counter function, the register is incremented in response to a 1-to-0 transition at the corresponding external input pin, T0 or T1. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a HIGH in one cycle and a LOW in the next cycle, the counter is incremented. Thus, it takes two machine cycles (24 oscillator periods) to recognize a 1- to-0 transition. There are no restrictions on the duty cycle of the external input signal, but to insure that a given level is sampled at least once before it changes, it should be held for at least one full machine cycle.

## 8 Single-chip 8-bit microcontroller

80CE558/83CE558/89CE558

Timer 0 and Timer 1 can be programmed independently to operate in one of four modes:

- Mode 0: 8-bit timer or 8-bit counter each with divide by-32 prescaler
- Mode 1: 16-bit time-interval or event counter
- Mode 2: 8-bit time-interval or event counter with automatic reload upon overflow
- Mode 3: -Timer 0: one 8-bit time-interval or event counter and one 8-bit time-interval counter  
-Timer 1: stopped

When Timer 0 is in Mode 3, Timer 1 can be

programmed to operate in Modes 0, 1 or 2 but cannot set an interrupt request flag or generate an interrupt. However the overflow from Timer 1 can be used to pulse the serial port baud-rate generator.

With a 16 MHz crystal, the counting frequency of these timer/counters is as follows:

- In the timer function, the timer is incremented at a frequency of 1,33 MHz - a division by 12 of the oscillator frequency
- 0 Hz to an upper limit of 0.66 MHz (1/24 of the oscillator frequency) when programmed for external inputs

Both internal and external inputs can be gated to the counter by a second external source for directly measuring pulse durations.

When configured as a counter, the register is incremented on every falling edge on the corresponding input pin, T0 or T1. The incremented register value can be read earliest during the second machine cycle after that one, during which the incrementing pulse occurred.

The counters are started and stopped under software control. Each one sets its interrupt request flag when it overflows from all HIGHS to all LOWs (or automatic reload value), with the exception of mode 3 as previously described.

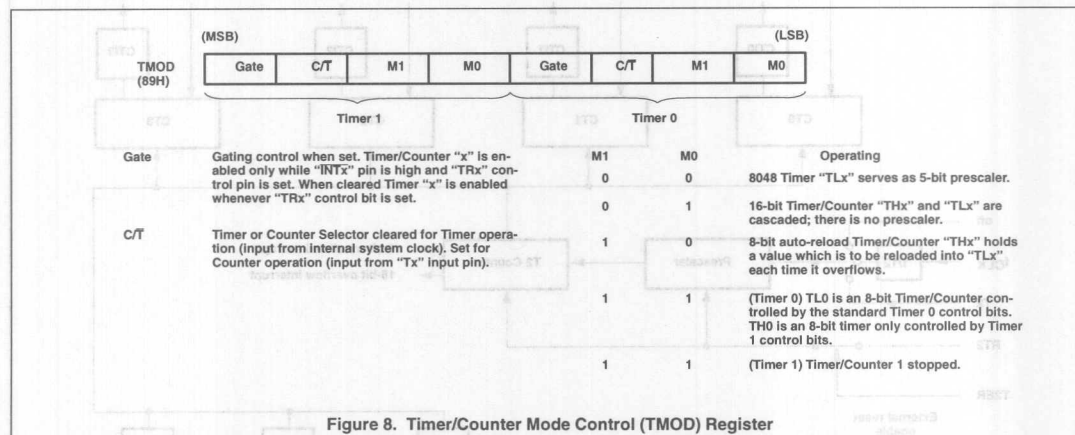


Figure 8. Timer/Counter Mode Control (TMOD) Register

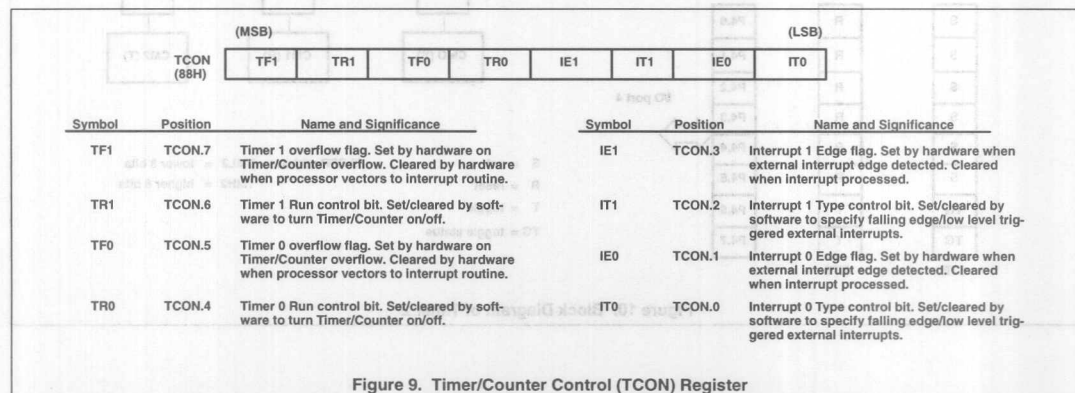


Figure 9. Timer/Counter Control (TCON) Register



## 8 Single-chip 8-bit microcontroller

## 80CE558/83CE558/89CE558

### 1.6.2 Timer T2

Timer T2 is a 16 bit timer/counter which has capture and compare facilities:

The operational diagram is shown in Figure 8.

The 16 bit timer/counter is clocked via a prescaler with a programmable division factor of 1, 2, 4 or 8. The input of the prescaler is clocked with 1/12 of the clock frequency, or by an external source connected to the T2 input, or it is switched off. The maximum repetition rate of the external clock source is  $f_{CLK}/12$ , twice that of Timer 0 and Timer 1. The prescaler is incremented on a rising

edge. It is cleared if its division factor or its input source is changed, or if the timer/counter is reset (see also Figure 11: TM2CON). T2 is readable 'on the fly'; without any extra read latches; this means that software precautions have to be taken against misinterpretation at overflow from least to most significant byte while T2 is being read. T2 is not loadable and is reset by the RST signal or at the positive edge of the input signal RT2, if enabled. In the Idle mode the timer/counter and prescaler are reset and halted.

T2 is connected to four 16-bit Capture

Registers: CT0, CT1, CT2 and CT3. A rising or falling edge on the inputs CT0I, CT1I, CT2I or CT3I (alternative function of Port 1) results in loading the contents of T2 into the respective Capture Registers and an interrupt request.

Using the Capture Register CTCON (see Figure 12), these inputs may invoke capture and interrupt request on a positive, a negative edge or on both edges. If neither a positive nor a negative edge is selected for capture input, no capture or interrupt request can be generated by this input.

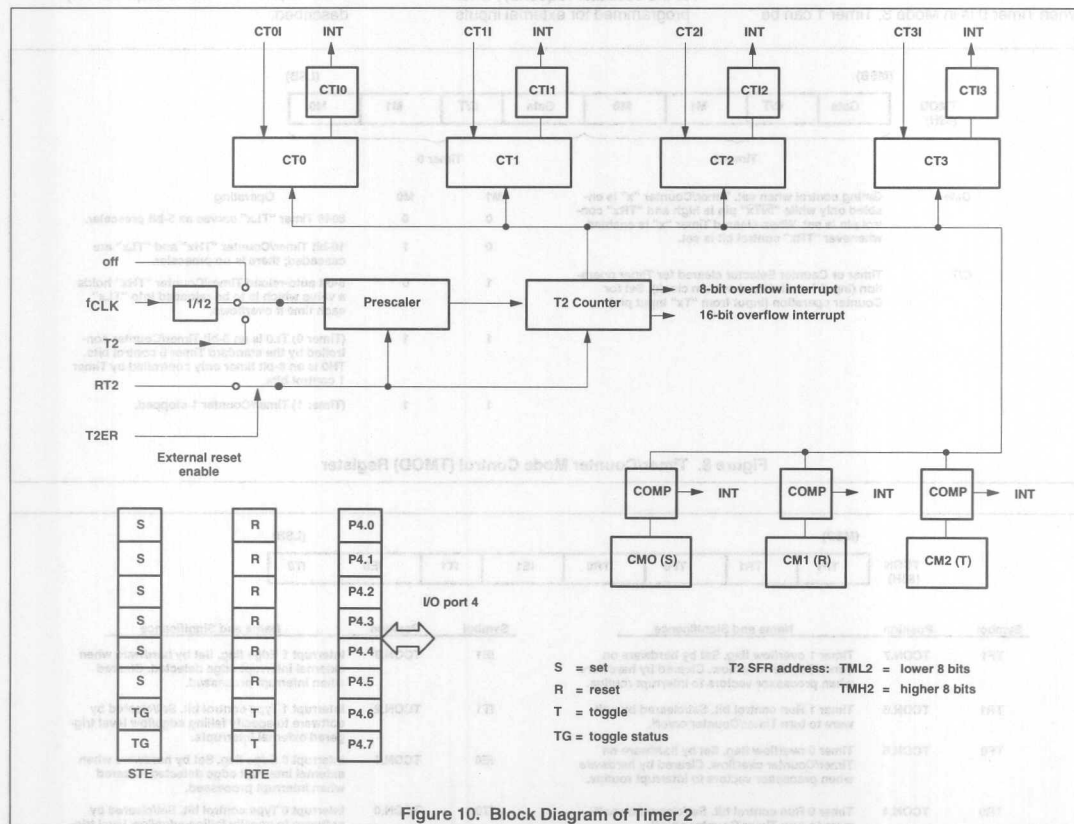


Figure 10. Block Diagram of Timer 2

## Single-chip 8-bit microcontroller

80CE558/83CE558/89CE558

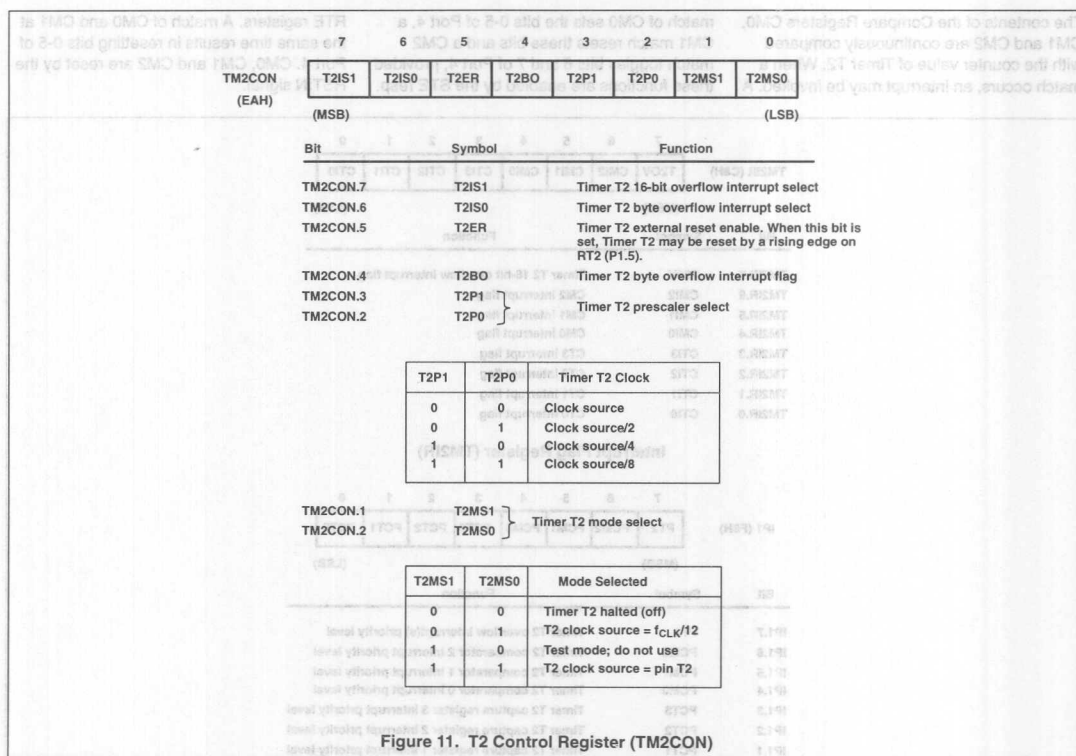


Figure 11. T2 Control Register (TM2CON)

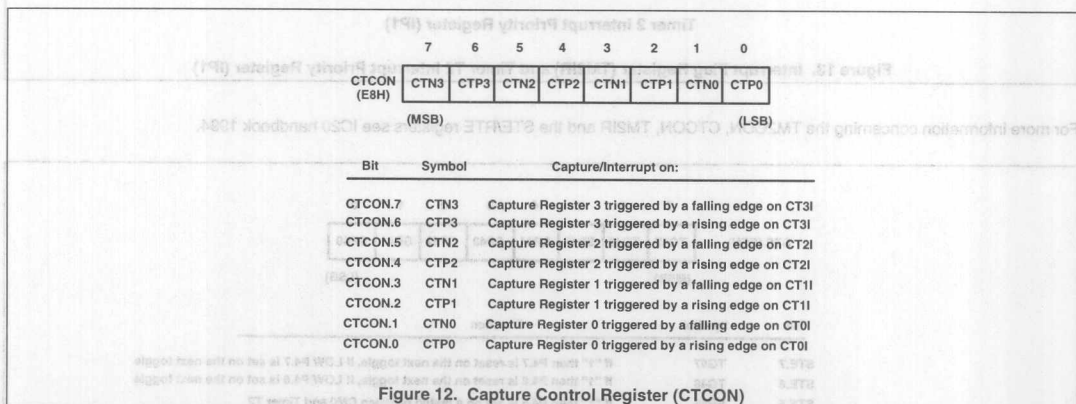


Figure 12. Capture Control Register (CTCON)

## Single-chip 8-bit microcontroller

80CE558/83CE558/89CE558

The contents of the Compare Registers CM0, CM1 and CM2 are continuously compared with the counter value of Timer T2. When a match occurs, an interrupt may be invoked. A

match of CM0 sets the bits 0-5 of Port 4, a CM1 match resets these bits and a CM2 match toggles bits 6 and 7 of Port 4, provided these functions are enabled by the STE resp.

RTE registers. A match of CM0 and CM1 at the same time results in resetting bits 0-5 of Port 4. CM0, CM1 and CM2 are reset by the RSTIN signal.

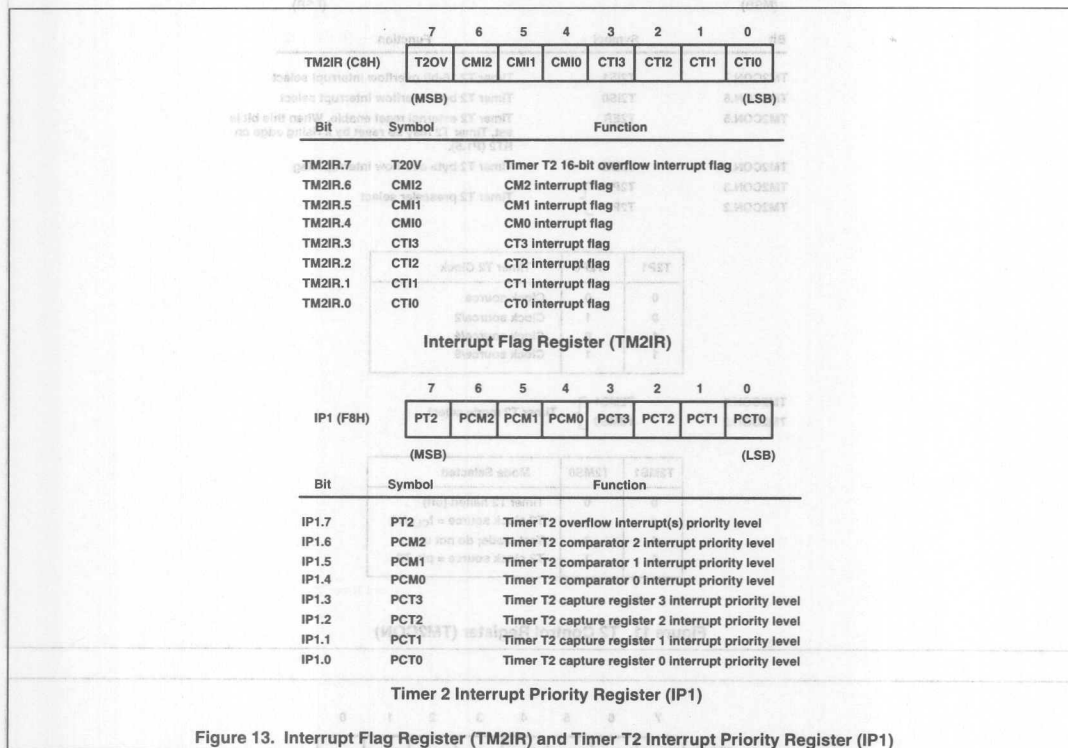


Figure 13. Interrupt Flag Register (TM2IR) and Timer T2 Interrupt Priority Register (IP1)

For more information concerning the TM2CON, CTCON, TM2IR and the STE/RTE registers see IC20 handbook 1994.

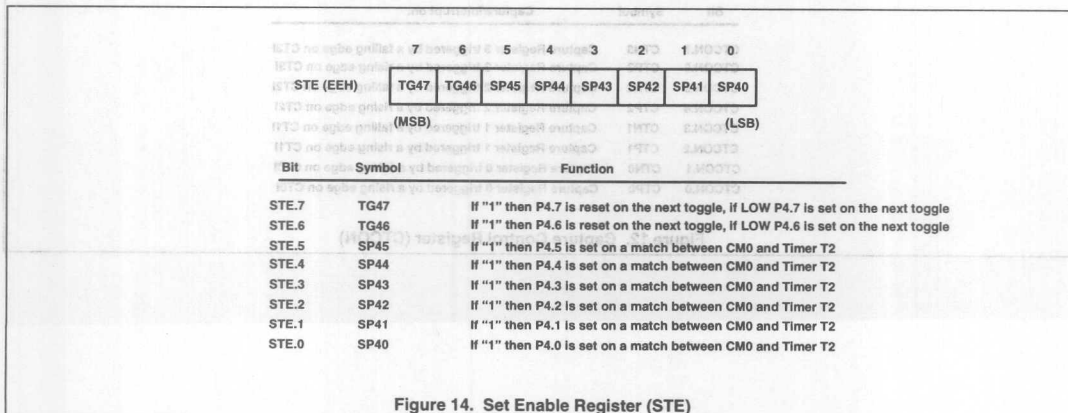


Figure 14. Set Enable Register (STE)

## Single-chip 8-bit microcontroller

## 80CE558/83CE558/89CE558

Port 4 can be read and written by software without affecting the toggle, set and reset signals. At a byte overflow of the least significant byte, or at a 16-bit overflow of the timer/counter, an interrupt sharing the same interrupt vector is requested. Either one or both of these overflows can be programmed to request an interrupt.

All interrupt flags must be reset by software.

### 1.7 Watchdog Timer (T3) (see Figure 16)

In addition to Timer T2 and the standard timers, a watchdog timer consisting of an 11-bit prescaler and an 8-bit timer is also incorporated.

The timer is incremented every 1,5 ms,

derived from the oscillator frequency of 16 MHz by the following:

$$f_{\text{timer}} = \frac{f_{\text{CLK}}}{12 \times 2048}$$

When a timer overflow occurs, the microcontroller - but not the PLL - is reset and a reset output pulse is generated at pin RSTOUT.

To prevent a system reset the timer must be reloaded in time by the application software. If the processor suffers a hardware/software malfunction, the software will fail to reload the timer. This failure will produce a reset upon overflow thus preventing the processor running out of control. The watchdog timer can only be reloaded if the condition flag

WLE = PCON.4 has been previously set by software.

At the moment the counter is loaded the condition flag is automatically cleared.

The time interval between the timer's reloading and the occurrence of a reset depends on the reloaded value. For example, this may range from 1,5 ms to 0,375 s when using an oscillator frequency of 16 MHz.

In the Idle state the watchdog timer and reset circuitry remain active.

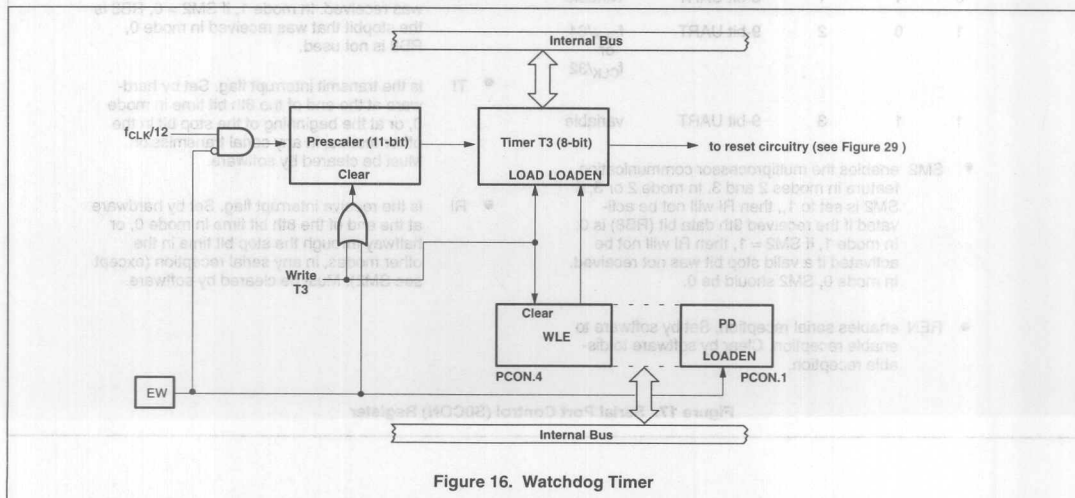
The watchdog timer is controlled by the watchdog enable pin (EW). A LOW level enables the watchdog timer and disables the Power-down mode. A HIGH level disables the watchdog timer and enables the Power-down mode.

Bit	Symbol	Function
7	TP47	TP47
6	TP46	TP46
5	RP45	RP45
4	RP44	RP44
3	RP43	RP43
2	RP42	RP42
1	RP41	RP41
0	RP40	RP40

(MSB) (LSB)

RTE (EFH)

Figure 15. Reset/Toggle Enable Register (RTE)



## 8 Single-chip 8-bit microcontroller

## 80CE558/83CE558/89CE558

## 1.8 Serial I/O

The 8xCE558 is equipped with two independent serial ports: SIO0 and SIO1. SIO0 is the full duplex UART port, identical to the PCB80C51 serial port. SIO1 is an I<sup>2</sup>C-bus serial I/O interface with byte oriented master and slave functions.

## 1.8.1 SIO0 (UART)

SIO 0 is a full duplex serial I/O port - it can transmit and receive simultaneously. This serial port is also receive-buffered. It can commence reception of a second byte before the previously received byte has been read from the receive register. If, however, the first byte has still not been read by the time reception of the second byte is complete, one of the bytes will be lost. The SIO0 receive and transmit registers are both accessed via the S0BUF special function register. Writing to S0BUF loads the transmit register, and reading S0BUF accesses to a physically separate receive register. SIO0 can operate in 4 modes:

**Mode 0:** Serial data is transmitted and received through RXD. TXD outputs the shift clock. 8 data bits

are transmitted/received (LSB first). The baud rate is fixed at 1/12 of the oscillator frequency.

**Mode 0:** 10 bits are transmitted via TXD or received through RXD: a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit is put into RB8 (S0CON special function register). The baud rate is variable.

**Mode 0:** 11 bits are transmitted through TXD or received through RXD: a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On transmit, the 9th data bit (TB8 in S0CON) can be assigned the value of 0 or 1. With nominal software, TB8 can be the parity bit (P in PSW). During a receive, the 9th data bit is stored in RB8 (S0CON), and the stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 of the oscillator frequency.

**Mode 0:** 11 bits are transmitted through TXD or received through RXD: a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and

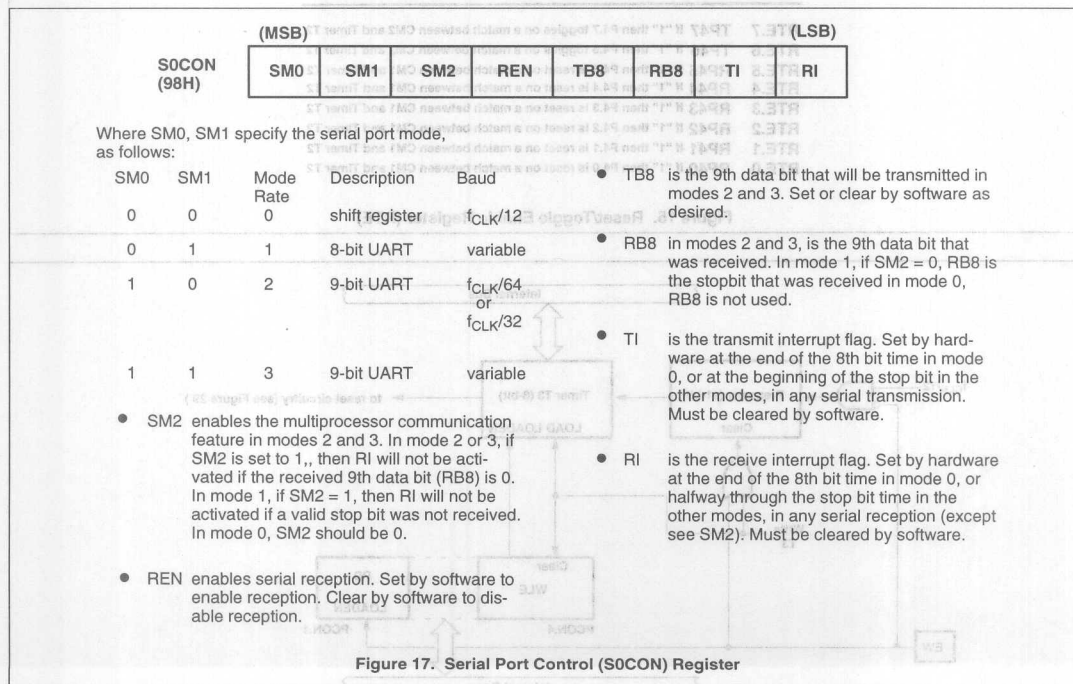
a stop bit (1). Mode 3 is the same as Mode 2 except the baud rate, which is variable in Mode 3.

In all four modes, transmission is initiated by any instruction that writes to the S0BUF function register. Reception is initiated in Mode 0 when RI = 0 and REN = 1. In the other three modes, reception is initiated by the incoming start bit provided that REN = 1.

Modes 2 and 3 are provided for multiprocessor communications: In these modes, 9 data bits are received with the 9th bit written to RB8. The 9th bit is followed by the stop bit. The port can be programmed so that with receiving the stop bit, the serial port interrupt will be activated if, and only if, RB8 = 1.

This feature is enabled by setting bit SM2 in S0CON. This feature may be used in multiprocessor systems.

For more information about how to use the UART in combination with the registers S0CON, PCON, IEN0, S0BUF and Timer register refer to the 80C51 Data Handbook IC20 1992.





## Single-chip 8-bit microcontroller

## 80CE558/83CE558/89CE558

### 1.8.2 SIO1 (I<sup>2</sup>C-bus Interface)

The I<sup>2</sup>C-bus is a simple bidirectional 2-wire bus for efficient inter-IC data exchange. Out-standing features of the I<sup>2</sup>C-bus are:

- Only two bus lines are required: a serial clock line (SCL) and a serial data line (SDA)
- Each device connected to the bus is software addressable by a unique address
- Masters can operate as Master-transmitter or as Master-receiver
- It's a true multi-master bus including collision detection and arbitration to prevent data corruption if two or more masters simultaneously initiate data transfer
- Serial clock synchronization allows devices with different bit rates to communicate via the same serial bus
- ICs can be added to or removed from an I<sup>2</sup>C-bus system without affecting any other circuit on the bus
- Fault diagnostics and debugging are simple; malfunctions can be immediately traced

The on-chip I<sup>2</sup>C logic provides a serial interface that meets the I<sup>2</sup>C-bus specification, supporting all I<sup>2</sup>C-bus modes of operation, they are:

- Master transmitter
- Master receiver
- Slave transmitter
- Slave receiver

The SIO1 logic performs a byte oriented data transport, clock generation, address recognition and bus control arbitration are all controlled by hardware. Via two pins the external I<sup>2</sup>C-bus is interfaced to the SIO1 logic:

SCL serial clock I/O and SDA (serial data I/O), (see Special Function Register bit S1CON.6/ENS1 for enabling the SIO1 logic).

The SIO1 logic handles byte transfer autonomously. It keeps track of the serial transfers, and a status register (S1STA) reflects the status of SIO1 and the I<sup>2</sup>C-bus.

Via the following four Special Function Registers the CPU interfaces to the I<sup>2</sup>C logic.

**S1CON** control register. Bit addressable by the CPU

**S1STA** status register whose contents may be used as a vector to service routines.

**S1DAT** data shift register. The data byte is stable as long as S1CON.3/SI=1.

**S1ADR** slave address register. It's LSB enables/ disables general call address recognition.

The Control Register, S1CON:  
The CPU can read from and write to this 8-bit, directly addressable SFR. Two bits are affected by the SIO1 hardware: the SI bit is set when a serial interrupt is requested, and the STO bit is cleared when a STOP

condition is present on the I<sup>2</sup>C bus. The STO bit is also cleared when ENS1 = "0".

## S1CON (D8H)

7	6	5	4	3	2	1	0
CR2	ENS1	STA	STO	SI	AA	CR1	CR0

### ENS1, the SIO1 Enable Bit

ENS1 = "0": When ENS1 is "0", the SDA and SCL outputs are in a high impedance state. SDA and SCL input signals are ignored, SIO1 is in the "not addressed" slave state, and the STO bit in S1CON is forced to "0". No other bits are affected.

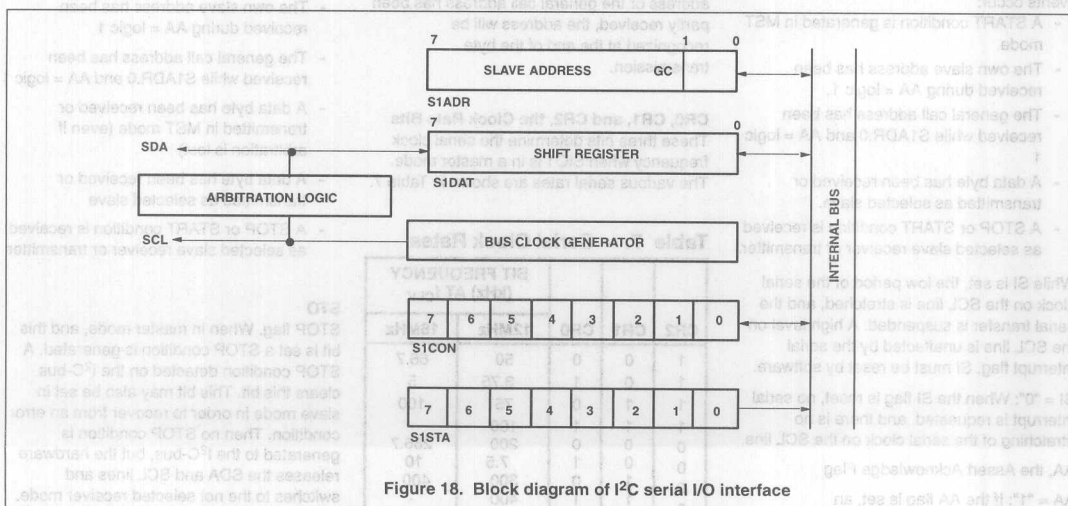
ENS1 = "1": When ENS1 is "1", SIO1 is enabled.

In the following text, it is assumed that  $\text{ENS1} = "1"$

ENS1 should not be used to temporarily release SIO1 from the I<sup>2</sup>C-bus since, when ENS1 is reset, the I<sup>2</sup>C bus status is lost. The AA flag should be used instead (see description of the AA flag in the following text).

### STA, the START Flag

STA = "1": When the STA bit is set to enter a master mode, the SIO1 hardware checks the status of the I2C bus and generates a START condition if the bus is free. If the bus is not free, then SIO1 waits for a STOP condition (which will free the bus) and generates a START condition after a delay of a half clock period of the internal serial clock generator.



**Figure 18. Block diagram of I<sup>2</sup>C serial I/O interface**

## Single-chip 8-bit microcontroller

## 80CE558/83CE558/89CE558

If STA is set while SIO1 is already in a master mode and one or more bytes are transmitted or received, SIO1 transmits a repeated START condition. STA may be set at any time. STA may also be set when SIO1 is an addressed slave.

STA = "0": When the STA bit is reset, no START condition or repeated START condition will be generated.

**STO, the STOP Flag**

STO = "1": When the STO bit is set while SIO1 is in a master mode, a STOP condition is transmitted to the I<sup>2</sup>C bus. When the STOP condition is detected on the bus, the SIO1 hardware clears the STO flag. In a slave mode, the STO flag may be set to recover from an error condition. In this case, no STOP condition is transmitted to the I<sup>2</sup>C bus. However, the SIO1 hardware behaves as if a STOP condition has been received and switches to the defined "not addressed" slave receiver mode. The STO flag is automatically cleared by hardware.

If the STA and STO bits are both set, the a STOP condition is transmitted to the I<sup>2</sup>C bus if SIO1 is in a master mode (in a slave mode, SIO1 generates an internal STOP condition which is not transmitted). SIO1 then transmits a START condition.

STO = "0": When the STO bit is reset, no STOP condition will be generated.

**SI, the Serial Interrupt Flag**

SI = "1": When the SI flag is set, then, if the EA and ES1 (interrupt enable register) bits are also set, a serial interrupt is requested. SI is set by hardware when one of the following events occur:

- A START condition is generated in MST mode.
- The own slave address has been received during AA = logic 1.
- The general call address has been received while S1ADR.0 and AA = logic 1
- A data byte has been received or transmitted as selected slave.
- A STOP or START condition is received as selected slave receiver or transmitter.

While SI is set, the low period of the serial clock on the SCL line is stretched, and the serial transfer is suspended. A high level on the SCL line is unaffected by the serial interrupt flag. SI must be reset by software.

SI = "0": When the SI flag is reset, no serial interrupt is requested, and there is no stretching of the serial clock on the SCL line.

**AA, the Assert Acknowledge Flag**

AA = "1": If the AA flag is set, an acknowledge (low level to SDA) will be

returned during the acknowledge clock pulse on the SCL line when:

- The "own slave address" has been received
- The general call address has been received while the general call bit (GC) in S1ADR is set
- A data byte has been received while SIO1 is in the master receiver mode
- A data byte has been received while SIO1 is in the addressed slave receiver mode

AA = "0": If the AA flag is reset, a not acknowledge (high level to SDA) will be returned during the acknowledge clock pulse on SCL when:

- A data has been received while SIO1 is in the master receiver mode
- A data byte has been received while SIO1 is in the addressed slave receiver mode

When SIO1 is in the not addressed slave mode, its own slave address and the general call address are ignored. Consequently, no acknowledge is returned, and a serial interrupt is not requested. Thus, SIO1 can be temporarily released from the I<sup>2</sup>C bus while the bus status is monitored. While SIO1 is released from the bus, START and STOP conditions are detected, and serial data is shifted in. Address recognition can be resumed at any time by setting the AA flag. If the AA flag is set when the part's own slave address or the general call address has been partly received, the address will be recognized at the end of the byte transmission.

**CR0, CR1, and CR2, the Clock Rate Bits**

These three bits determine the serial clock frequency when SIO1 is in a master mode. The various serial rates are shown in Table 7.

**Table 7. Serial Clock Rates**

CR2	CR1	CR0	BIT FREQUENCY (kHz) AT f <sub>CLK</sub>	
			12MHz	16MHz
1	0	0	50	66.7
1	0	1	3.75	5
1	1	0	75	100
1	1	1	100	-
0	0	0	200	266.7
0	0	1	7.5	10
0	1	0	300	400
0	1	1	400	-

The frequencies shown in Table 7 are unimportant when SIO1 is in a slave mode. In the slave modes, SIO1 will automatically synchronize with any clock frequency up to 100kHz.

Except from the bit rate selection (see Table 7) and the timing of the SCL and SDA signals (see AC electrical characteristics in section 10 of this data sheet) the SIO circuit is the same as described in detail in the 8xC552 section of this handbook.

**AA**

Assert acknowledge bit. When this bit is set, an acknowledge is returned after any one of the following conditions:

- Own slave address is received
- General call address is received (S1ADR.0 = logic 1)
- A data byte is received, while the device is programmed to be a master receiver
- A data byte is received, while the device is a selected slave receiver

When the bit is reset, no acknowledge is returned. Consequently, no interrupt is requested when the own address or general is received.

**SI**

SIO1 interrupt flag. This is set, and an interrupt request is generated, after any of the following events occur:

- A START condition is generated in MST mode
- The own slave address has been received during AA = logic 1
- The general call address has been received while S1ADR.0 and AA = logic 1
- A data byte has been received or transmitted in MST mode (even if arbitration is lost)
- A data byte has been received or transmitted as selected slave
- A STOP or START condition is received as selected slave receiver or transmitter

**STO**

STOP flag. When in master mode, and this bit is set a STOP condition is generated. A STOP condition detected on the I<sup>2</sup>C-bus clears this bit. This bit may also be set in slave mode in order to recover from an error condition. Then no STOP condition is generated to the I<sup>2</sup>C-bus, but the hardware releases the SDA and SCL lines and switches to the not selected receiver mode. The STOP flag is cleared by the hardware.

## 8 Single-chip 8-bit microcontroller

## 80CE558/83CE558/89CE558

**STA** START flag. When this bit is set in slave mode, the hardware checks the I<sup>2</sup>C-bus and generates a START condition if the bus is free or after the bus becomes free. If the device operates in master mode it will generate a repeated START condition.

**ENS1**

0 = Serial I/O Disabled and reset. SDA and SCL outputs are open drain.

1 = Serial I/O Enabled.

**Serial status register S1STA** (S1STA is a read only register)

**S1STA (D9H)**

7	6	5	4	3	2	1	0
SC4	SC3	SC2	SC1	SC0	0	0	0

S1STA.3 - S1STA.7 hold a status code. S1STA.0 - S1STA.2 are held LOW. The contents of the status register may be used as a vector to a service routine. This optimizes the response time of the software and consequently that of the I<sup>2</sup>C-bus.

**Abbreviations used:**

SLA : 7-bit slave address

R : Read bit

W : Write bit

ACK : Acknowledgement (acknowledge bit = logic 0)

ACK : Not acknowledgement (acknowledge bit = logic 1)

DATA : 8b-t data byte to or from I<sup>2</sup>C-bus

MST : Master

SLV : Slave

TRX : Transmitter

REC : Receiver

The following is a list of the status codes:

**MST/TRX mode****S1STA value**

- 08H - A START condition has been transmitted
- 10H - A repeated START condition has been transmitted
- 18H - SLA and W have been transmitted, ACK has been received
- 20H - SLA and W have been transmitted, ACK received
- 28H - DATA and S1DAT has been transmitted, ACK received
- 30H - DATA and S1DAT has been transmitted, ACK received
- 38H - Arbitration lost in SLA, R/W or DATA

**MST/REC mode****S1STA value**

- 38H - Arbitration lost while returning ACK
- 40H - SLA and R have been transmitted, ACK received
- 48H - SLA and R have been transmitted, ACK received
- 50H - DATA has been received, ACK returned
- 58H - DATA has been received, ACK returned

**SLV/REC mode****S1STA value**

- 60H - Own SLA and W have been received, ACK returned
- 68H - Arbitration lost in SLA, R/W as MST. Own SLA and W have been received, ACK returned
- 70H - General CALL has been received, ACK returned
- 78H - Arbitration lost in SLA, R/W as MST. General call has been received
- 80H - Previously addressed with own SLA. DATA byte received, ACK returned
- 88H - Previously addressed with own SLA. DATA byte received, ACK returned
- 90H - Previously addressed with general call. DATA byte has been received, ACK has been returned
- 98H - Previously addressed with general call. DATA byte has been received, ACK has been returned
- A0H - A STOP condition or repeated START condition has been received while still addressed as SLV/REC or SLV/TRX.

**SLV/TRX mode****S1STA value**

- A8H - Own SLA and R have been received, ACK returned
- B0H - Arbitration lost in SLA, R/W as MST. Own SLA and R have been received, ACK returned
- B8H - DATA byte has been transmitted, ACK returned
- C0H - DATA byte has been transmitted, ACK returned
- C8H - Last DATA byte has been transmitted (AA = logic 0), ACK received

**Miscellaneous****S1STA value**

- 00H - Bus error during MST mode or selected (AA = logic 0), ACK received

**The data shift register S1DAT****S1DAT (DAH)**

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

This register contains the serial data to be transmitted or data which has been received. Bit 7 is transmitted or received first; i.e. data is shifted from right to left.

**The address register S1ADR****S1ADR (DBH)**

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

S1ADR.0, GC : 0 = general call address is not recognized  
1 = general call address recognized

S1ADR.7 - 1 : own slave address

This 8-bit register may be loaded with the 7-bit slave address to which the controller will respond when programmed as a slave receiver/transmitter. The LSB (GC) is used to determine whether the general call address is recognized.

**1.9 Interrupt System**

External events and the real-time-driven on-chip peripherals require service by the CPU asynchronously to the execution of any particular section of code. To tie the asynchronous activities of these functions to normal program execution a multiple-source, two-priority-level, nested interrupt system is provided. Interrupt response time in a single-interrupt system is in the range from 2.25us to 6.75 us when using a 16 MHz crystal. The latency time depends on the sequence of instructions executed directly after an interrupt request.

The 8xCE558 acknowledges interrupt requests from 15 sources as follows (see Figure 19):

- UART serial I/O port receive/transmit interrupt
- I<sup>2</sup>C-bus interface serial I/O interrupt
- ADC autoscan completion interrupt
- 'Seconds' timer interrupt SEC (ORed with INT1)
- INT0 and INT1 external interrupts
- Timer 0 and Timer 1 internal timer/counter interrupts
- Timer 2 internal timer/counter byte and/or 16-bit overflow, 3 compare and 4 capture interrupts (or 4 additional external interrupts). (1)

## 8 Single-chip 8-bit microcontroller

80CE558/83CE558/89CE558

The External Interrupts INT0 and INT1 can each be either level-activated or transition-activated, depending on bits IT0 and IT1 in register TCON. The flags that actually generate these interrupts are bits IEO and IE1 in TCON. When an external interrupt is generated, the corresponding request flag is cleared by the hardware when the service routine is vectored to only if the interrupt was transition-activated. If the interrupt was level-activated then the interrupt request flag remains set until the external interrupt pin INTx goes high. Consequently the external source has to hold the request active until the requested interrupt is actually generated. Then it has to deactivate the request before the interrupt service routine is completed, or else another interrupt will be generated. As these external interrupts are active LOW a "wire-ORing" of several interrupt sources to one input pin allows expansion.

The Timer 0 and Timer 1 Interrupts are generated by TF0 and TF1, which are set by a rollover in their respective timer/counter register (except for Timer 0 in Mode 3 of the serial interface). When a Timer interrupt is generated, the flag that generated it is cleared by the on-chip hardware when the service routine is vectored to.

The eight Timer/Counter T2 Interrupt sources are: 4 capture Interrupts (\*), 3 compare interrupts and an overflow interrupt. The appropriate interrupt request flags must be cleared by software.

The UART Serial Port Interrupt is generated by the logical OR of RI and TI. Neither of these flags is cleared by hardware. The service routine will normally have to determine whether it was RI or TI that generated the interrupt, and the bit will have to be cleared by software.

The I<sup>2</sup>C Interrupt is generated by bit SI in register S1CON. This flag has to be cleared by software.

The ADC Interrupt is generated by bit ADINT, which is set when of all selected analog inputs to be scanned, the conversion is finished. ADINT must be cleared by software. It cannot be set by software.

All of the bits that generate interrupts can be

set or cleared by software, with the same result as though it had been set or cleared by hardware (except the ADC interrupt request flag ADINT, which cannot be set by software). That is, interrupts can be generated or pending interrupts can be cancelled in software.

The Interrupts X0, T0, X1, T1, SEC, S0 and S1 are capable to terminate the Idle Mode.

#### Interrupt Enable Registers

Each interrupt source can be individually enabled or disabled by setting or clearing a bit in the interrupt enable special function registers IEN0 and IEN1. All interrupt sources can also be globally enabled or disabled by setting or clearing bit EA in IEN0. The interrupt enable registers are described in Figures 20 and 21.

#### Interrupt Priority Structure

Each interrupt source can be assigned one of two priority levels. Interrupt priority levels are defined by the interrupt priority special function registers IP0 and IP1. IP0 and IP1 are described in Figures 22 and 23.

Interrupt priority levels are as follows:

"0"—low priority

"1"—high priority

A low priority interrupt may be interrupted by a high priority interrupt. A high priority interrupt cannot be interrupted by any other interrupt source. If two requests of different priority occur simultaneously, the high priority level request is serviced. If requests of the same priority are received simultaneously, an internal polling sequence determines which request is serviced. Thus, within each priority level, there is a second priority structure determined by the polling sequence. This second priority structure is shown in Table 8.

#### Interrupt Handling

The interrupt sources are sampled at S5P2 of every machine cycle. The samples are polled during the following machine cycle. If one of the flags was in a set condition at S5P2 of the previous machine cycle, the polling cycle will find it and the interrupt system will generate an LCALL to the appropriate service routine, provided this hardware-generated LCALL is not blocked by any of the following conditions:

1. An interrupt of higher or equal priority level is already in progress.
2. The current machine cycle is not the final cycle in the execution of the instruction in progress. (No interrupt request will be serviced until the instruction in progress is completed.)
3. The instruction in progress is RETI or any access to the interrupt priority or interrupt enable registers. (No interrupt will be serviced after RETI or after a read or write to IP0, IP1, IEO, or IE1 until at least one other instruction has been subsequently executed.)

The polling cycle is repeated with every machine cycle, and the values polled are the values present at S5P2 of the previous machine cycle. Note that if an interrupt flag is active but is not being responded to because of one of the above conditions, and if the flag is inactive when the blocking condition is removed, then the blocked interrupt will not be serviced. Thus, the fact that the interrupt flag was once active but not serviced is not remembered. Every polling cycle is new.

The processor acknowledges an interrupt request by executing a hardware-generated LCALL to the appropriate service routine. In some cases it also clears the flag which generated the interrupt, and in others it does not. It clears the Timer 0, Timer 1, and external interrupt flags. An external interrupt flag (IE0 or IE1) is cleared only if it was transition-activated. All other interrupt flags

are not cleared by hardware and must be cleared by the software. The LCALL pushes the contents of the program counter on to the stack (but it does not save the PSW) and reloads the PC with an address that depends on the source of the interrupt being vectored to as shown in Table 9.

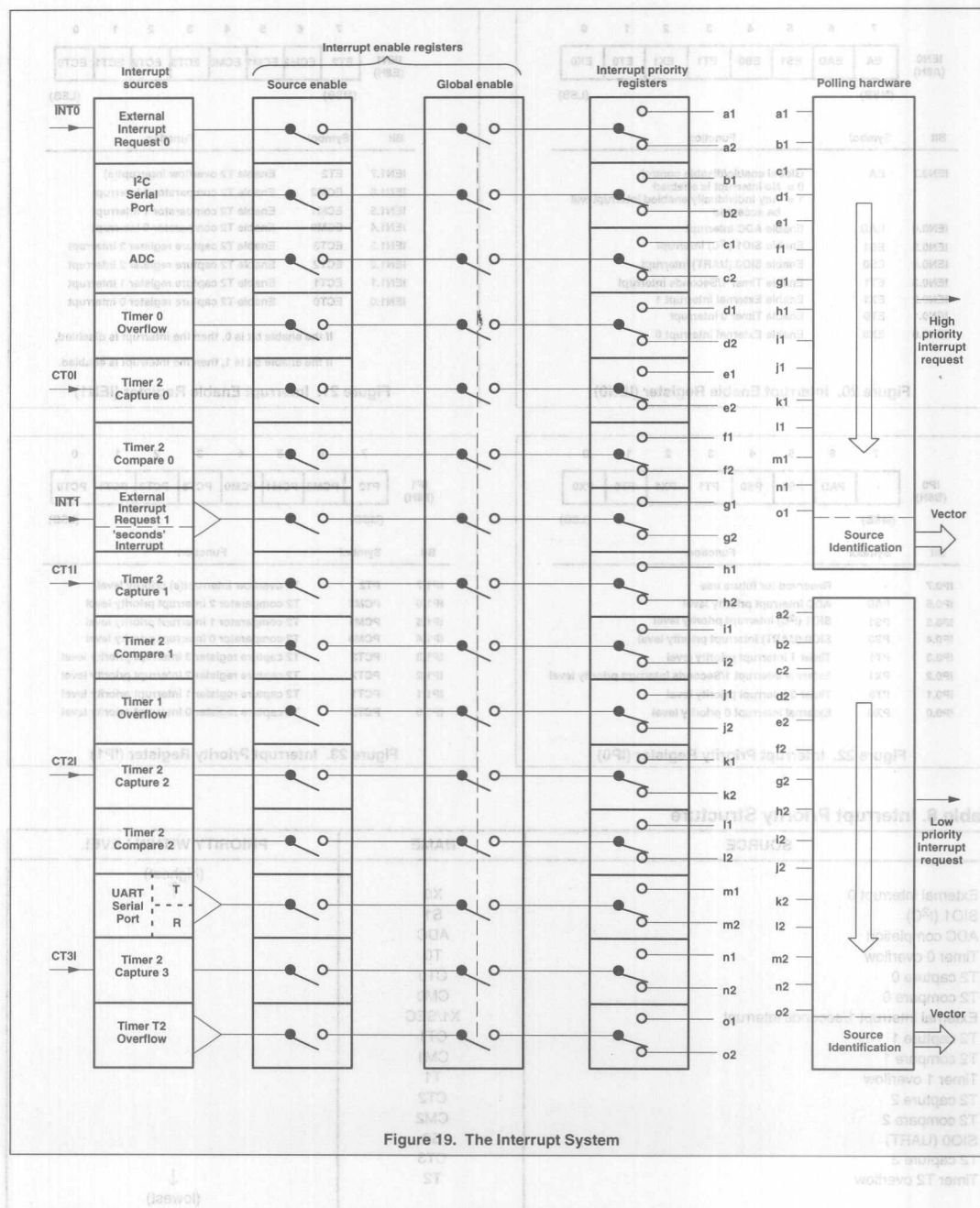
Execution proceeds from the vector address until the RETI instruction is encountered. The RETI instruction clears the "priority level active" flip-flop that was set when this interrupt was acknowledged. It then pops the top two bytes from the stack and reloads the program counter. Execution of the interrupted program continues from where it was interrupted.

**Note 1):** If a capture register is unused and it's contents is of no interest, then the corresponding input pin CTn/P1.n (n: 0...3) may be used as a (configurable) positive and/or negative edge triggered additional external interrupt input (INT2, INT3, INT4, INT5).



## Single-chip 8-bit microcontroller

## 80CE558/83CE558/89CE558





80CE558/83CE558/89CE558

80CE558/83CE558/89CE558

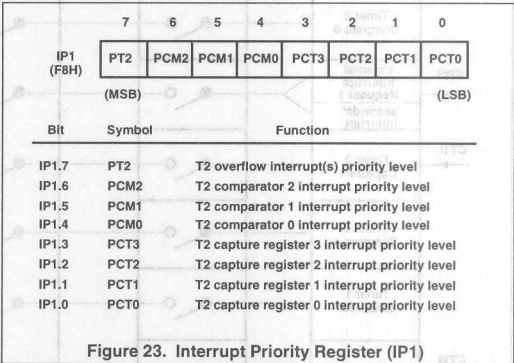
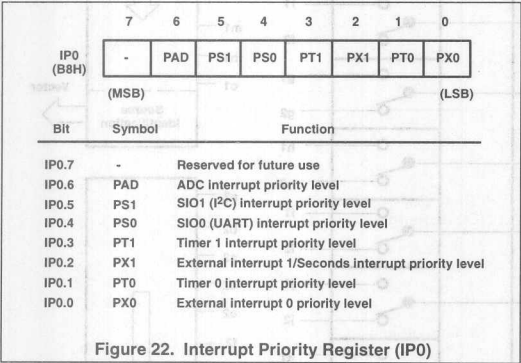
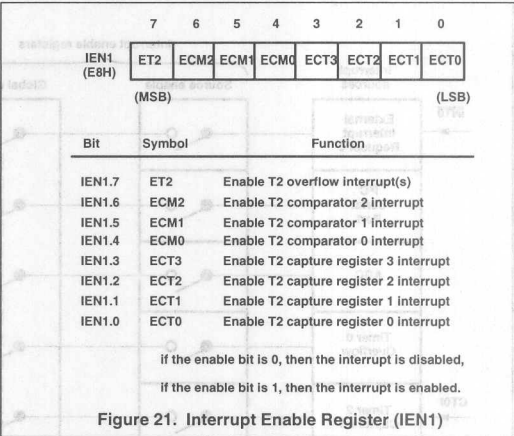
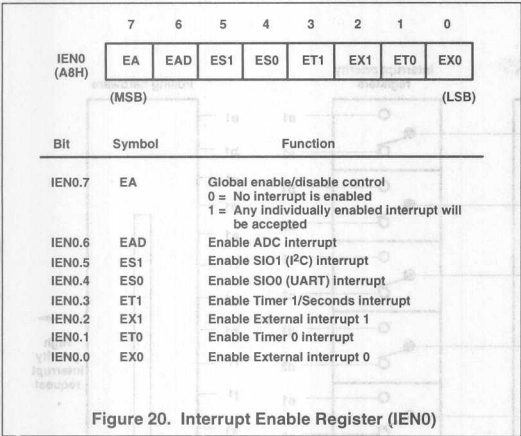


Table 8. Interrupt Priority Structure

SOURCE	NAME	PRIORITY WITHIN LEVEL
External interrupt 0	X0	(highest)
SIO1 (I <sup>2</sup> C)	S1	↑
ADC completion	ADC	
Timer 0 overflow	T0	
T2 capture 0	CT0	
T2 compare 0	CM0	
External interrupt 1/seconds interrupt	X1/SEC	
T2 capture 1	CT1	
T2 compare 1	CM1	
Timer 1 overflow	T1	
T2 capture 2	CT2	
T2 compare 2	CM2	
SIO0 (UART)	S0	
T2 capture 3	CT3	
Timer T2 overflow	T2	↓ (lowest)

Single-chip 8-bit microcontroller

80CE558/83CE558/89CE558

Table 9. Interrupt Vector Addresses

SOURCE	NAME	VECTOR ADDRESS
External interrupt 0	X0	0003H
Timer 0 overflow	T0	000BH
External interrupt 1/seconds interrupt	X1/SEC	0013H
Timer 1 overflow	T1	001BH
SIO0 (UART)	S0	0023H
SIO1 (I <sup>2</sup> C)	S1	002BH
T2 capture 0	CT0	0033H
T2 capture 1	CT1	003BH
T2 capture 2	CT2	0043H
T2 capture 3	CT3	004BH
ADC completion	ADC	0053H
T2 compare 0	CM0	005BH
T2 compare 1	CM1	0063H
T2 compare 2	CM2	006BH
T2 overflow	T2	0073H

Bit	Symbol	Function
7	PCON (87H)	
6	SMOD	Double Baud rate bit. When set to logic 1 the baud rate is doubled when the serial port SIO0 is being used in modes 1, 2, or 3.
5	ARD	AUX-RAM disable bit. When set to a 1 the internal 768-byte AUX-RAM is disabled, so that all MOVX-Instructions access the external data memory - as it is with the standard PCB80C51.
4	RFI	Reduced radio frequency interference bit. When set to a 1 the toggling of ALE pin is prohibited. This bit is cleared on RESET (see also sections Features (EMC) and Pinning).
3	WLE	Watchdog load enable. This flag must be set by software prior to loading timer T3 (watchdog timer). It is cleared when timer T3 is loaded.
2	GF1	General-purpose flag bit
1	GF0	General-purpose flag bit
0	PD	Power-down bit. Setting this bit activates the power-down mode. It can only be set if input EW is high.
	IDL	Idle mode bit. Setting this bit activates the idle mode.

If logic 1s are written to PD and IDL at the same time, PD takes precedence.

Figure 24. Power Control Register (PCON)

## Single-chip 8-bit microcontroller

## 80CE558/83CE558/89CE558

## 1.10 Power reduction modes

Two software-selectable modes of reduced power consumption are implemented. These are the Idle Mode and the Power-down Mode.

Idle Mode operation permits the interrupt, serial ports and timer blocks T0, T1 and T3 to function while the CPU is halted. The following functions are switched off when the microcontroller enters the Idle Mode:

- CPU (halted)
- Timer 2 (stopped and reset)
- PWM0, PWM1 (reset, output = HIGH)
- ADC (aborted if conversion in progress)

The following functions remain active during Idle Mode. These functions may generate an interrupt or reset and thus terminate the Idle Mode:

- Timer 0, Timer 1, Timer 3 (Watchdog timer)
- UART
- I<sup>2</sup>C
- External interrupt
- Second Timer

In Power-down Mode the system clock is halted. If the PLL oscillator is selected (SELXTAL1 = 0) and the RUN32 bit is set, the 32 kHz oscillator keeps running, otherwise its stopped. If the HF-oscillator (SELXTAL1 = 1) is selected, it is frozen after setting the bit PD in the PCON register.

## 1.10.1 Power Control Register

The modes Idle and Power-down are activated by software via the Special Function Register PCON. Its hardware address is 87h. PCON is not bit addressable. The reset value of PCON is (00000000).

## 1.10.2 Idle Mode

The instruction that sets PCON.0 is the last instruction executed in the normal operating mode before Idle mode is activated. Once in the Idle mode, the CPU status is preserved in its entirety: the Stack Pointer, Program Counter, Program Status Word, Accumulator, RAM and all other registers maintain their data during Idle mode.

The status of external pins during Idle mode is shown in Table 10.

There are three ways to terminate the Idle mode:

Activation of any enabled interrupt X0, T0, X1, SEC, T1, S0 or S1 will cause PCON.0 to be cleared by hardware terminating Idle mode. The interrupt is serviced, and following return from interrupt instruction RETI, the next instruction to be executed will be the one which follows the instruction that wrote a logic 1 to PCON.0.

The flag bits GF0 and GF1 may be used to determine whether the interrupt was received during normal execution or during Idle mode. For example, the instruction that writes to PCON.0 can also set or clear one or both flag bits. When Idle mode is terminated by an interrupt, the service routine can examine the status of the flag bits.

The second way of terminating the Idle mode is with an external hardware reset. Since the oscillator is still running, the hardware reset is required to be active for two machine cycles (24 HF oscillator periods) to complete the reset operation if the HF oscillator is selected.

Table 10. External Pin Status During Idle and Power-Down Modes

MODE	MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3	PORT 4	SCL/SDA	PWM0/PWM1
Idle	Internal	1	1	data	data	data	data	data	operative (1)	HIGH
Idle	External	1	1	high - Z	data	Address	data	data	operative (1)	HIGH
Power-down	Internal	0	0	data	data	data	data	data	high-Z	HIGH
Power-down	External	0	0	high - Z	data	data	data	data	high-Z	HIGH

## NOTE:

1. In Idle Mode SCL and SDA can be active as outputs only if SIO1 is enabled; if SIO1 is disabled (S1CON.6/ENS1 = 0) these pins are in a high-impedance state.

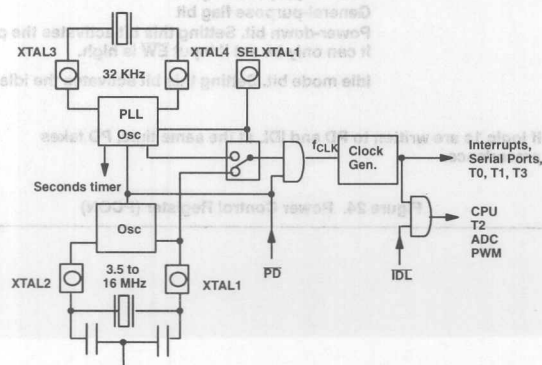


Figure 25. Idle and Power Down Hardware

## Single-chip 8-bit microcontroller

## 80CE558/83CE558/89CE558

When the PLL oscillator is selected a hardware reset of  $\geq 1 \mu\text{sec}$  is required and the microcontroller restarts within 63 msec after the reset has finished.

The third way of terminating the Idle mode is by internal watchdog reset. The microcontroller restarts after 3 machine cycles in all cases.

### 1.10.3 Power-down mode

The instruction that sets PCON.1 is the last executed prior to going into the Power-down mode. Once in Power-down mode, the HF oscillator is stopped. The 32 KHz oscillator may stay running. The content of the on-chip RAM and the Special Function Registers are not saved. Note that the Power-down mode can not be entered when the watchdog has been enabled.

The Power-down mode can be terminated by an external RESET in the same way as in the 80C51 (RAM is saved, SFRs are cleared due to RESET) or in addition by any one of the external interrupts, INT0, INT1 or Seconds interrupt. (see description Wake-up from Power-down mode)

The status of the external pins during Power-down mode is shown in Table 10. If the Power-down mode is activated while in external program memory, the port data that is held in the Special Function Register P2 is restored to Port 2. If the data is a logic1, the port pin is held HIGH during the Power-down mode by the strong pull-up transistor P1 (see Figure 5)

### 1.10.4 Wake-up from Power-down Mode

The Power-down mode of the 8xCE558 can also be terminated by any one of the three

enabled interrupts, INT0, INT1 or Seconds interrupt.

A termination with these interrupts do not affect the internal data memory and does not affect the Special Function Registers. This gives the possibility to exit Power-down without changing the port output levels. To terminate the Power-down mode with an external interrupt, INT0 or INT1 must be switched to be level-sensitive and must be enabled. The external interrupt input signal INT0 or INT1 must be kept LOW till the oscillator has restarted and stabilized (see Figure 26) in order to prevent any interrupt priority problems during wake-up. A Seconds interrupt will terminate the Power-down mode if enabled and INT1 is level sensitive.

Wake-up time is 60 ms in this case. The priority of the desired wake-up interrupt should be higher than the priorities of all other enabled interrupt sources. The instruction following the one that put the device into the Power-down mode will be the first one which will be executed after the wake-up.

## 1.11 Oscillator circuits

The input signal SELXTAL1 selects for XTAL1, 2 oscillator (standard 80C51) or the PLL oscillator. The not selected oscillator is halted and its XTAL pin must be not connected.

### 1.11.1 XTAL1, 2 Oscillator circuit (standard 80C51)

The oscillator circuit of the 8xCE558 is a single-stage inverting amplifier in a Pierce oscillator configuration. The circuitry between the XTAL1 and XTAL2 is basically an inverter biased to the transfer point. Either a crystal or

ceramic resonator can be used as the feedback element to complete the oscillator circuitry. Both are operated in parallel resonance. XTAL1 is the high gain amplifier input, and XTAL2 is the output (see Figure 27). To drive the 8xCE558 externally, XTAL1 is driven from an external source and XTAL2 left open-circuit (see Figure 28).

### 1.11.2 XTAL3, 4 circuitry, PLL oscillator

See appendix 1, "Specification PLL oscillator for 80C51 derivatives."

## 1.12 Reset Circuitry

The reset input pin RSTIN is connected to a Schmitt trigger for noise reduction (see Figure 29). Is the HF-oscillator selected a reset is accomplished by holding the RSTIN pin HIGH for at least 2 machine cycles (24 system clock periods). Is the PLL-oscillator selected the RSTIN-pulse must have a width of  $1 \mu\text{s}$  at least, independent of the 32 KHz-oscillator is running or not (see PLL description). The CPU responds by executing an internal reset. The RSTOUT pin represents the signal resetting the CPU and can be used to reset peripheral devices.

The RSTOUT level also could be high due to a Watchdog timer overflow.

The length of the output pulse from T3 is 3 machine cycles. A pulse of such short duration is necessary in order to recover from a processor or system fault as fast as possible.

During reset, ALE and PSEN output a HIGH level. In order to perform a correct reset, this level must not be affected by external elements.

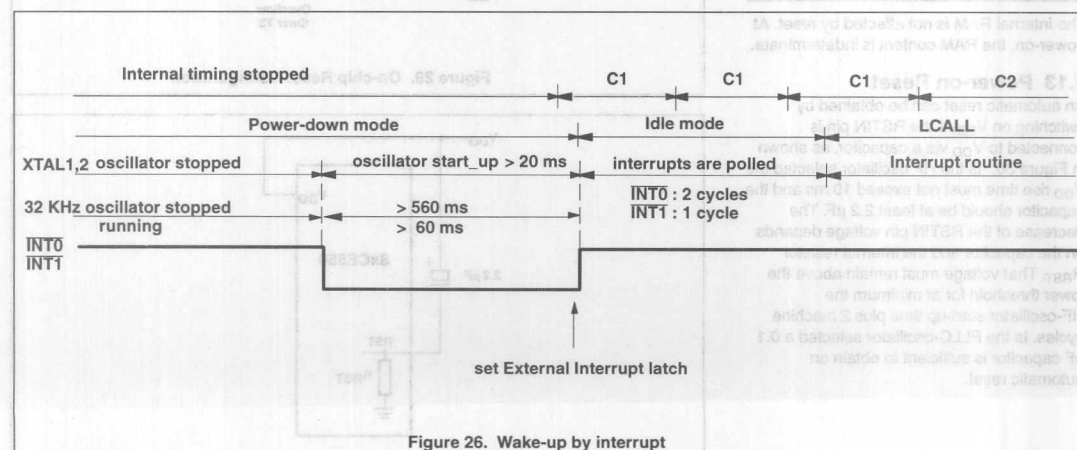


Figure 26. Wake-up by interrupt

## Single-chip 8-bit microcontroller

## 80CE558/83CE558/89CE558

A reset leaves the internal registers as follows:

REGISTER	CONTENT
ACC	0000
ADCON	xx00
ADRESH	xxxx
ADPSS	0000
ADRSLO-7	xxxx
B	0000
CML0-CML2	0000
CMH0-CMH2	0000
CTCON	0000
CTL0-CTL3	xxxx
CTH0-CTH3	xxxx
DPL	0000
DPH	0000
FMCON	0000
IEN0	0000
IEN1	0000
IP0	x000
IP1	0000
PCON	0000
PLLCON	0000
PSW	0000
PWM0	0000
PWM1	0000
PWMP	0000
P0-P4	1111
P5	xxxx
RTE	0000
S0BUF	xxxx
S0CON	0000
S1ADR	0000
S1CON	0000
S1DAT	0000
S1STA	1111
SP	0000
STE	1100
TCON	0000
TH0, TH1	0000
TMH2	0000
TL0, TL1	0000
TML2	0000
TMOD	0000
TM2CON	0000
TM2IR	0000
T3	0000
XRAMP	xxxx

The internal RAM is not affected by reset. At power-on, the RAM content is indeterminate.

### 1.13 Power-on Reset

An automatic reset can be obtained by switching on  $V_{DD}$ , if the RSTIN pin is connected to  $V_{DD}$  via a capacitor, as shown in Figure 30. Is the HF oscillator selected the  $V_{DD}$  rise time must not exceed 10 ms and the capacitor should be at least 2.2  $\mu\text{F}$ . The decrease of the RSTIN pin voltage depends on the capacitor and the internal resistor  $R_{RST}$ . That voltage must remain above the lower threshold for at minimum the HF-oscillator start-up time plus 2 machine cycles. Is the PLLC-oscillator selected a 0.1 nF capacitor is sufficient to obtain an automatic reset.

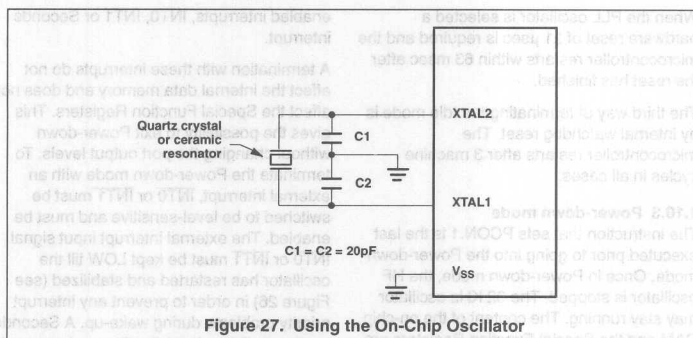


Figure 27. Using the On-Chip Oscillator

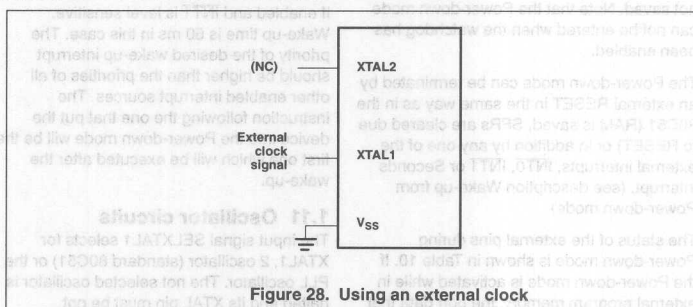


Figure 28. Using an external clock

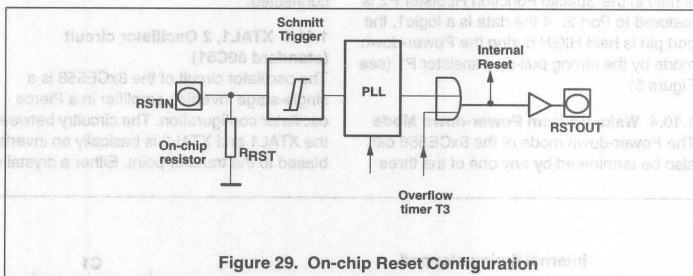


Figure 29. On-chip Reset Configuration

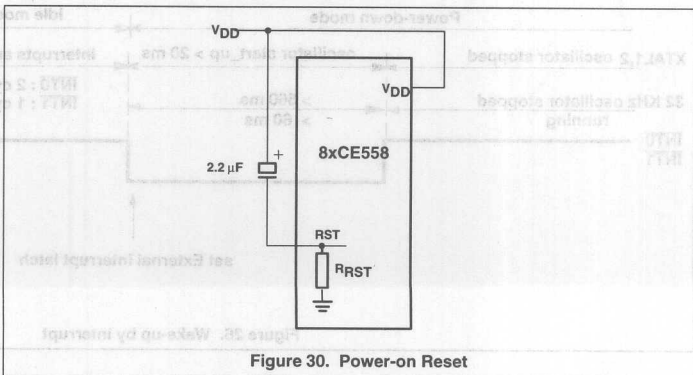


Figure 30. Power-on Reset



## 8. Single-chip 8-bit microcontroller

80CE558/83CE558/89CE558

### 2.0 INSTRUCTION SET

The 8xCE558 uses the powerful instruction set of the PCB80C51. It consists of 49 single-byte, 45 two-byte and 17 three-byte instructions. Using a 16 MHz quartz, 64 of the instructions are executed in 0.75  $\mu$ s, 45 in 1.5  $\mu$ s and the multiply, divide instructions in 3  $\mu$ s.

A summary of the instruction set is given in Table 11.

The 8xCE558 has additional Special Function Registers to control the on-chip peripherals.

### 2.1 Addressing Modes

Most instructions have a "destination, source" field that specifies the data type, addressing modes and operands involved. For all these instructions, except for MOVs, the

destination operand is also the source operand (e.g. ADD A,R7).

There are five kinds of addressing modes:

- Register Addressing
    - R0 - R7 (4 banks)
    - A,B,C (bit), AB (2 bytes), DPTR (double byte)
  - Direct Addressing
    - lower 128 bytes of internal Main RAM (including the 4 R0-R7 register banks)
    - Special Function Registers 1
    - 28 bits in a subset of the internal Main RAM
    - 128 bits in a subset of the Special Function Registers
  - Register-Indirect Addressing
    - internal Main RAM (@R0, @R1, @SP [PUSH/POP])
    - internal Auxiliary RAM (@R0, @R1, @DPTR)
    - external Data Memory (@R0, @R1, @DPTR)
  - Immediate Addressing
    - Program Memory (in-code 8 bit or 16 bit constant)
  - Base-Register-plus Index-Register-Indirect Addressing
    - Program Memory look-up table (@DPTR+A, @PC+A)
- The first three addressing modes are usable for destination operands.

### 80C51 FAMILY INSTRUCTION SET

Table 11. 80C51 Instruction Set Summary

Interrupt Response Time: Refer to Hardware Description Chapter.

### Instructions that Affect Flag Settings<sup>(1)</sup>

Instruction	Flag			Instruction	Flag		
	C	OV	AC		C	OV	AC
ADD	X	X	X	CLR C	0		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C,bit	X		
MUL	0	X		ANL C,bit	X		
DIV	0	X		ANL C,bit	X		
DA	X			ORL C,bit	X		
RRC	X			MOV C,bit	X		
RLC	X			CJNE	X		
SETB C	1						

<sup>(1)</sup>Note that operations on SFR byte address 208 or bit addresses 209-215 (i.e., the PSW or bits in the PSW) will also affect flag settings.

### Notes on instruction set and addressing modes:

Rn	Register R7-R0 of the currently selected Register Bank.
direct	8-bit internal data location's address. This could be an Internal Data RAM location (0-127) or a SFR [i.e., I/O port, control register, status register, etc. (128-255)].
@Ri	8-bit internal data RAM location addressed indirectly through register R1 or R0 of the actual register bank..
#data	8-bit constant included in the instruction.
#data 16	16-bit constant included in the instruction
addr 16	16-bit destination address. Used by LCALL and LJMP. A branch can be anywhere within the 64k-byte Program Memory address space.
addr 11	11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2k-byte page of program memory as the first byte of the following instruction.
rel	Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is -128 to +127 bytes relative to first byte of the following instruction.
bit	Direct Addressed bit in Internal Data RAM or Special Function Register.

All mnemonics copyrighted © Intel Corporation 1980

## 88 Single-chip 8-bit microcontroller

## 80CE558/83CE558/89CE558

Table 11. 80C51 Instruction Set Summary (Continued)

MNEMONIC		DESCRIPTION	BYTE	CYCLES
<b>ARITHMETIC OPERATIONS</b>				
ADD	A,Rn	Add register to Accumulator	1	1
ADD	A,direct	Add direct byte to Accumulator	2	1
ADD	A,@Ri	Add indirect RAM to Accumulator	1	1
ADD	A,#data	Add immediate data to Accumulator	2	1
ADDC	A,Rn	Add register to Accumulator with carry	1	1
ADDC	A,direct	Add direct byte to Accumulator with carry	2	1
ADDC	A,@Ri	Add indirect RAM to Accumulator with carry	1	1
ADDC	A,#data	Add immediate data to A <sub>CC</sub> with carry	2	1
SUBB	A,Rn	Subtract Register from A <sub>CC</sub> with borrow	1	1
SUBB	A,direct	Subtract direct byte from A <sub>CC</sub> with borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from A <sub>CC</sub> with borrow	1	1
SUBB	A,#data	Subtract immediate data from A <sub>CC</sub> with borrow	2	1
INC	A	Increment Accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Increment indirect RAM	1	1
DEC	A	Decrement Accumulator	1	1
DEC	Rn	Decrement Register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
INC	DPTR	Increment Data Pointer	1	2
MUL	AB	Multiply A and B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal Adjust Accumulator	1	1
<b>LOGICAL OPERATIONS</b>				
ANL	A,Rn	AND Register to Accumulator	1	1
ANL	A,direct	AND direct byte to Accumulator	2	1
ANL	A,@Ri	AND indirect RAM to Accumulator	1	1
ANL	A,#data	AND immediate data to Accumulator	2	1
ANL	direct,A	AND Accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to Accumulator	1	1
ORL	A,direct	OR direct byte to Accumulator	2	1
ORL	A,@Ri	OR indirect RAM to Accumulator	1	1
ORL	A,#data	OR immediate data to Accumulator	2	1
ORL	direct,A	OR Accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive-OR register to Accumulator	1	1
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	1

All mnemonics copyrighted © Intel Corporation 1980

## Single-chip 8-bit microcontroller

## 80CE558/83CE558/89CE558

Table 11. 80C51 Instruction Set Summary (Continued)

	MNEMONIC	BYTE	DESCRIPTION	BYTE	CYCLES
LOGICAL OPERATIONS (Continued)					
XRL	A,@Ri	1	Exclusive-OR indirect RAM to Accumulator	1	1
XRL	A,#data	2	Exclusive-OR immediate data to Accumulator	2	1
XRL	direct,A	1	Exclusive-OR Accumulator to direct byte	2	1
XRL	direct,#data	2	Exclusive-OR immediate data to direct byte	3	2
CLR	A	1	Clear Accumulator	1	1
CPL	A	1	Complement Accumulator	1	1
RL	A	1	Rotate Accumulator left	1	1
RLC	A	1	Rotate Accumulator left through the carry	1	1
RR	A	1	Rotate Accumulator right	1	1
RRC	A	1	Rotate Accumulator right through the carry	1	1
SWAP	A	1	Swap nibbles within the Accumulator	1	1
DATA TRANSFER					
MOV	A,Rn	1	Move register to Accumulator	1	1
MOV	A,direct	2	Move direct byte to Accumulator	2	1
MOV	A,@Ri	1	Move indirect RAM to Accumulator	1	1
MOV	A,#data	2	Move immediate data to Accumulator	2	1
MOV	Rn,A	1	Move Accumulator to register	1	1
MOV	Rn,direct	2	Move direct byte to register	2	2
MOV	Rn,#data	2	Move immediate data to register	2	1
MOV	direct,A	2	Move Accumulator to direct byte	2	1
MOV	direct,Rn	2	Move register to direct byte	2	2
MOV	direct,direct	1	Move direct byte to direct	3	2
MOV	direct,@Ri	2	Move indirect RAM to direct byte	2	2
MOV	direct,#data	3	Move immediate data to direct byte	3	2
MOV	@Ri,A	1	Move Accumulator to indirect RAM	1	1
MOV	@Ri,direct	2	Move direct byte to indirect RAM	2	2
MOV	@Ri,#data	2	Move immediate data to indirect RAM	2	1
MOV	DPTR,#data16	3	Load Data Pointer with a 16-bit constant	3	2
MOVC	A,@A+DPTR	1	Move Code byte relative to DPTR to Acc	1	2
MOVC	A,@A+PC	1	Move Code byte relative to PC to Acc	1	2
MOVX	A,@Ri	1	Move external RAM (8-bit addr) to Acc	1	2
MOVX	A@DPTR	1	Move external RAM (16-bit addr) to Acc	1	2
MOVX	A,@Ri,A	1	Move Acc to external RAM (8-bit addr)	1	2
MOVX	@DPTR,A	1	Move Acc to external RAM (16-bit addr)	1	2
PUSH	direct	2	Push direct byte onto stack	2	2
POP	direct	1	Pop direct byte from stack	2	2
XCH	A,Rn	1	Exchange register with Accumulator	1	1
XCH	A,direct	2	Exchange direct byte with Accumulator	2	1
XCH	A,@Ri	1	Exchange indirect RAM with Accumulator	1	1
XCHD	A,@Ri	1	Exchange low-order digit indirect RAM with Acc	1	1

All mnemonics copyrighted © Intel Corporation 1980

## 8 Single-chip 8-bit microcontroller

80CE558/83CE558/89CE558

Table 11. 80C51 Instruction Set Summary (Continued)

INSTRUCTION	OPERANDS	DESCRIPTION	BYTE	CYCLES
<b>BOOLEAN VARIABLE MANIPULATION</b>				
CLR	C	Clear carry	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set carry	1	1
SETB	bit	Set direct bit	2	1
CPL	C	Complement carry	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to carry	2	2
ANL	C,/bit	AND complement of direct bit to carry	2	2
ORL	C,bit	OR direct bit to carry	2	2
ORL	C,/bit	OR complement of direct bit to carry	2	2
MOV	C,bit	Move direct bit to carry	2	1
MOV	bit,C	Move carry to direct bit	2	2
JC	rel	Jump if carry is set	2	2
JNC	rel	Jump if carry not set	2	2
JB	rel	Jump if direct bit is set	2	2
JNB	rel	Jump if direct bit is not set	2	2
JBC	bit,rel	Jump if direct bit is set and clear bit	3	2
<b>PROGRAM BRANCHING</b>				
ACALL	addr11	Absolute subroutine call	2	2
LCALL	addr16	Long subroutine call	3	2
RET		Return from subroutine	1	2
RETI		Return from interrupt	1	2
AJMP	addr11	Absolute jump	2	2
LJMP	addr16	Long jump	3	2
SJMP	rel	Short jump (relative addr)	2	2
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if Accumulator is zero	2	2
JNZ	rel	Jump if Accumulator is not zero	2	2
CJNE	A,direct,rel	Compare direct byte to A <sub>CC</sub> and jump if not equal	3	2
CJNE	A,#data,rel	Compare immediate to A <sub>CC</sub> and jump if not equal	3	2
CJNE	Rn,#data,rel	Compare immediate to register and jump if not equal	3	2
CJNE	@Ri,#data,rel	Compare immediate to indirect and jump if not equal	3	2
DJNZ	Rn,rel	Decrement register and jump if not zero	2	2
DJNZ	direct,rel	Decrement direct byte and jump if not zero	3	2
NOP		No operation	1	1

All mnemonics copyrighted © Intel Corporation 1980

## Single-chip 8-bit microcontroller

## 80CE558/83CE558/89CE558

## 3.0 FLASH EEPROM

## 3.1 General

- 32 kbyte electrically erasable internal program memory with Block-and-Page-Erase option ("Flash Memory").
- Internal fixed boot ROM.
- Up to 32 kbyte external program memory in combination with the internal EEPROM (EA=1).
- Up to 64 kbyte external program memory if the internal program memory is switched off (EA=0).

The EEPROM can be read and written byte-wise. Full Erase, Block Erase, and Page erase will erase 32 kbyte, 256 bytes and 32 bytes respectively. In-circuit programming and out-of-circuit programming is possible. On-chip erase and write timing generation and on chip high voltage generation contribute to a user friendly interface.

## 3.2 Features

- Read: byte-wise
- Write: byte-wise within 2.5 ms. (previously erased by a page, block or full erase).
- Erase: Page Erase (32 bytes) within 5 ms. Block Erase (256 bytes) within 5 ms. Full Erase (32 kbyte) within 5 ms. Erased bytes contain FFH.
- Endurance: 100 erase and write cycles each byte at  $T_{amb} = 22^{\circ}\text{C}$
- Retention: 10 years
- Out-of-circuit programming: Parallel programming with 87C51 compatible hardware interface to programmer.
- In-circuit programming: Serial programming via RS232 interface under boot ROM program control. Auto baud rate selection. Intel Hex Object file Format.
- The user program can call routines in the boot ROM for erase, write and verify of the EEPROM.
- High programming voltage generation: on chip
- Zero point on-chip oscillator and timer to generate the write and erase time durations.
- Programmable security for the code in the EEPROM to prevent software piracy. The Security Byte is located in the highest address (7FFFH) of the EEPROM.

- Supply voltage monitoring circuit on-chip to prevent loss of information in the EEPROM during power-on and power-off.

## 3.3 Memory map

Figure 31 shows the memory map of the user program memory and the boot ROM. They are located in the same program address space. Two bits UBS1 and UBS0 of the EEPROM control special function register FMCON select between the two memory blocks.

## User program memory selection

If UBS1 and UBS0 are both 0, then the user program memory is mapped into the 64k program memory space and the boot ROM cannot be selected. This is the situation after a reset when PSEN and ALE have not been pulled down during reset. Program execution starts at 0000H in the internal EEPROM or in the external program memory dependent on the level of EA during reset.

## Boot ROM selection

If UBS1 and UBS0 are both '1' then the boot ROM is mapped into the 64k program memory space and the user program memory cannot be selected. This is the situation after a reset when during reset PSEN and EA are pulled down while ALE stay high. Program execution starts at 0000H of the boot ROM. The boot ROM size is 1 kbyte and its address map is repeated after each 1k addresses. Besides serial the in-circuit programming routines the Boot ROM contains the routines for erase write and verify of the EEPROM which can be called by the user program (LJMP/LCALL to the address space between 63k and 64k).

## Switching between user program memory and boot ROM

Switching between user program memory (internal or external) and boot ROM is possible if UBS1 and UBS0 are 0,1 or 1,0. Then in the program memory address space between 0 and 63k the user program memory is selected and in the memory space between 63k and 64k the boot ROM is selected.

To switch from user program memory to boot ROM first UBS0 must be set (UBS1 stay 0) and a jump or call instruction to a location >63k must be executed. At the moment of crossing the 63k address border by a call or jump instruction the switching between user and boot memory is performed without timing problems.

To switch from boot ROM to user program memory the boot program will be running between 63k and 64k. UBS1 is set and UBS0 is cleared and a jump or return instruction to

a location <63k must be executed. At the moment of crossing the 63k address border by a jump or return instruction the switching from boot ROM to user memory (internal or external) is performed. After crossing the 63k address border UBS1 is immediately cleared by hardware and the total 64k memory space is mapped as user program memory. By clearing UBS1 by hardware, no special requirements to the user program are necessary to do that after a boot routine or erase or write routine.

A small restriction for memory switching is that no memory switching is allowed from or to the address space between 63k and 64k of the user program memory because the UBS bits must stay 0 in this range and no 63k address crossing would take place. This restriction can be avoided if the memory switching is always done by a subroutine in the address range between 0 and 63k.

## Description

The user program code in the EEPROM is executed as in the standard 80C51 microcontroller. Erase and write cycles in the EEPROM are always performed under control of the boot program in the boot ROM in the address space between 63k and 64k. Address and data parameters are passed via DPTR and accumulator A respectively. During an erase or write cycle in the EEPROM no other access or program execution in the EEPROM is possible. All interrupts must be disabled when the user program calls a user routine in the boot ROM.

The boot routine for serial programming takes care of addressing, data transfer, blank-check, verify, high voltage control, error message, interrupt disabling/enabling and return to the user program memory. It also contains the serial communication routine.

The EEPROM control register FMCON is a special function register. It contains the control bits for verify, write, erase and boot ROM switching.

## FMCON (FBH)

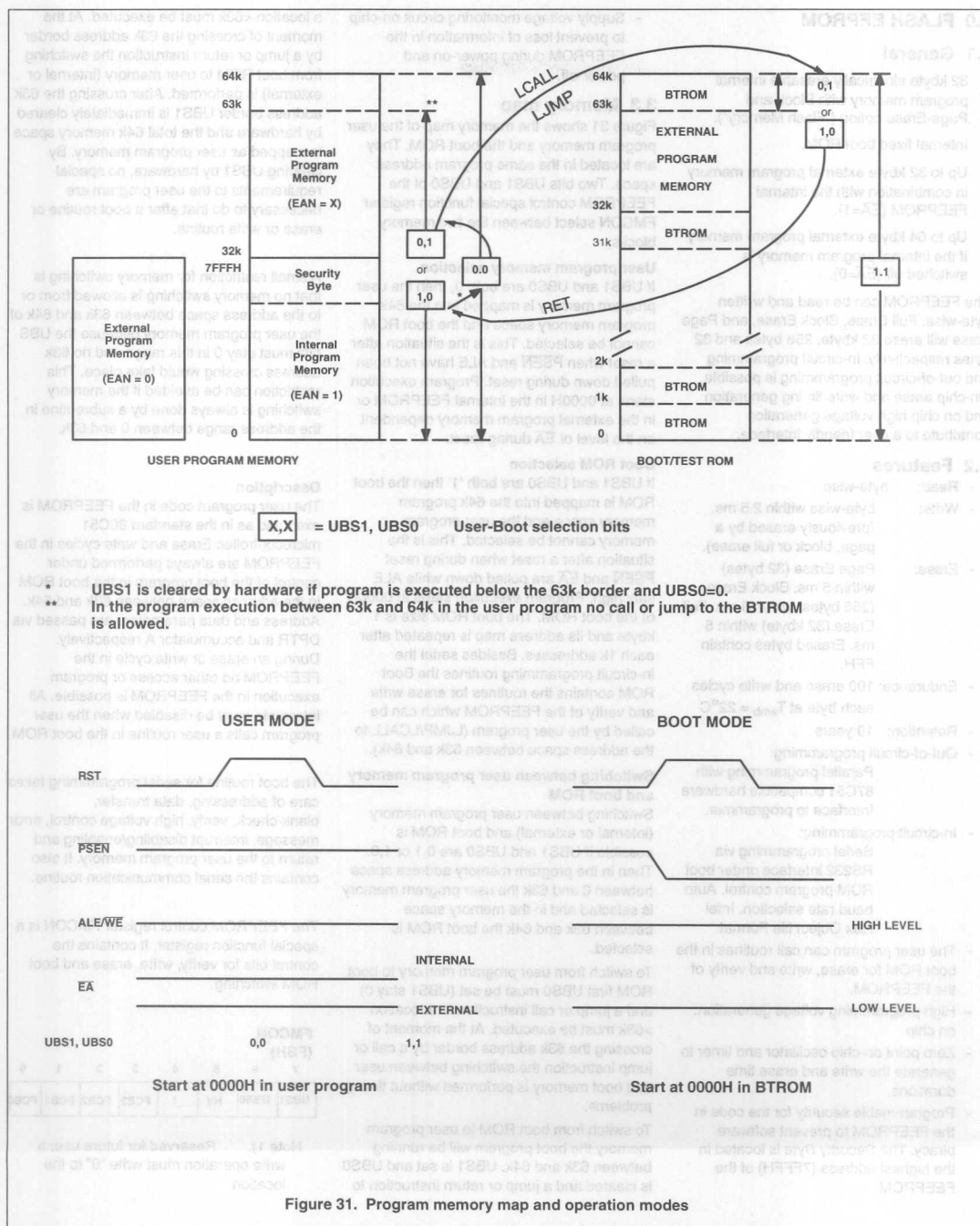
7	6	5	4	3	2	1	0
UBS1	UBS0	HV	_1	FCB3	FCB2	FCB1	FCB0

Note 1): Reserved for future use; a write operation must write "0" to the location



## 8 Single-chip 8-bit microcontroller

## 80CE558/83CE558/89CE558



## Single-chip 8-bit microcontroller

80CE558/83CE558/89CE558

UBS1	UBS0	User - Boot selection bits
0	0	User memory mapped from 0 to 64k
0	1	User memory mapped from 0 to 63k Boot ROM mapped from 63k to 64k
1	0	User memory mapped from 0 to 63k, but UBS1 bit cleared by hardware in this user address range. Boot ROM mapped from 63k to 64k.
1	1	Boot ROM mapped from 0 to 64k.

HV	High voltage indication bit. Read only. Is 1 as long as the high voltage for an erase or write operation is present.
----	--

FCB3	FCB2	FCB1	FCB0	Function Code Bits
0	0	0	0	Value after reset.
0	1	0	1	Byte Write or byte read (verify)
1	1	0	0	Page Erase (32 byte boundaries).
0	0	1	1	Block Erase (256 byte boundaries).
1	0	1	0	Full Erase (32k byte).

The four FCB bits are by write protected if the security feature is activated. Then only instructions in the internal program memory (EEPROM) are able to write FCB3-0, boot ROM and external program memory instructions cannot change FCB3-0 except the full erase code can be loaded.

The duration of a write or erase operation is determined by the EEPROM timer. This timer includes a zero point RC oscillator and cannot be controlled by software.

For calling a user routine in the Boot ROM first all interrupts must be disabled and the DPTR and A have to be loaded with the desired values. After setting UBS0 = 1 and UBS1 = 0 and selecting the function via FCB-bits the respective user routine will be called.

The table below lists the boot ROM user routines, which can be called by the user program. The content of FMCON, A and DPTR before calling is by (IN) and by (OUT)

after described return. The boot ROM user routines do not change other registers or Data memory.

X = don't care or not defined

V = verified byte (read back)

1) = 5 LSB's of DPTR are don't care

2) = 5 LSB's of DPTR are "0"

3) = 8 LSB's of DPTR are don't care

4) = 8 LSB's of DPTR contain 08H.

Example of user software (internal or external) that calls the Page Erase routine in the boot-ROM to erase a page in the EEPROM (32 bytes) starting at address location 1260H.

CLR EA

MOV DPTR, # 1260H ; Disable all interrupts

MOV FMCON, # 4CH ; Load page-address

LCALL 0FFAAH ; Call Page-Erase code

SETB EA ; Enable interrupts again

LCALL 0FFAAH ; Call Page-Erase routine

SETB EA ; Enable interrupts again

Example of user software (internal or external) that calls the Byte-Write routine in the boot-ROM to write the content of R5 into the EEPROM address location 1263H.

CLK EA ; Disable all interrupts

MOV DPTR, # 1263H ; Load byte address

MOV A, R5 ; Load byte to be written

MOV FMCON, # 45H ; Load byte-write code

LCALL 0FFADH ; Call byte-write routine

SETB EA ; Enable interrupts again

XRL A, R5 ; Compare the "read-back" byte

JNZ ; Jump if verify error

BOOT-ROM ROUTINE	CALL ADDRESS	FMCON (IN)	FMCON (OUT)	A (IN)	A (OUT)	DPTR (IN)	DPTR (OUT)
BYTE_READ	FFBAH	45H	15H	XXH	BYTE	BYTE ADDRESS	BYTE ADDRESS
BYTE_WRITE	FFADH	45H	15H	BYTE	BYTE (V)	BYTE ADDRESS	BYTE ADDRESS
PAGE_ERASE	FFAAH	46H	16H	XXH	08H	PAGE ADDRESS <sup>1)</sup>	PAGE ADDRESS <sup>2)</sup>
BLOCK_ERASE	FFA5H	43H	13H	XCXH	02H	BLOCK ADDRESS <sup>3)</sup>	BLOCK ADDRESS <sup>4)</sup>
FULL_ERASE	FFA0H	4AH	1AH	XXH	0AH	XXXXH	0018H

Single-chip 8-bit microcontroller

80CE558/83CE558/89CE558

3.4 Security

The security feature protects against software piracy and prevents that the content of the FEEPROM can be read undesirable. The Security Byte is located in the highest address location 7FFFH of the FEEPROM and is the same as all other bytes of the FEEPROM.

The Security Byte should be 01010000B to activate and 00H or FFH to deactivate the security feature. This security code is chosen in such a way that single bit failures will not change the security mode.

If the security feature is deactivated, then there are no access restrictions to the FEEPROM.

If the security feature is activated, then the external program memory has no access to the FEEPROM with the MOV instructions. Also bits FCB3-0 of FMCON cannot be written by external program code or boot ROM code. This prevents in-circuit programming and verification. Only the Full Erase code can be written to FCB0-3 of FMCON. At the end of a full erase operation the security feature is deactivated. Also parallel programming and verify is inhibited if the security feature is activated, only a full

erase is possible. Note that the security mode does not change immediately when the security code is written into the security byte 7FFFH, but after a reset or power-on. This allows the verification of the loaded code in the FEEPROM, including the Security Byte.

3.5 Parallel programming

Unlike standard EPROM programming, no high programming supply voltage must be applied to the EA pin and only one programming pulse must be applied to the ALE/WE pin. The parallel programming mode is entered with the steady signals RST=1, PSEN=0, EA=1 and SELTAL1 = 1. The XTAL1, 2 clock must have a frequency between 4 and 6MHz. The following table shows the logic levels for programming, erasing, verifying and read signature.

MODE	ALE/WE	P2.7	P2.6	P3.7	P3.6
Full erase		1	1	0	1
Program FEEPROM		1	0	1	1
Verify FEEPROM	0	0	1	1	1
Read signature	1	0	0	0	0

ALE/WE Write Enable signal (program/erase), active low  
P2.6, P2.7, P3.6, P3.7 output enable signal for verify/read modes, active low

Data and address bits:

P0.0-P0.7: D0 - D7 Program data input / verify or read data output

P1.0-P1.7: A0 - A7 Input low order address bits.

P2.0-P2.5, P3.4 : A8 - A14 Input high order address bits.

The 89CE558 contains two signature bytes that can be read and used by an EPROM programming system to identify the device. These bytes are read by the same procedure as for a normal verification of locations 30H and 31H, except that P3.6 and P3.7 need to be pulled to LOW.

ADDRESS	CONTENT	MEANING
30H	15H	Philips
31H	B5H	89CE558

The 89CE558 has a Security Byte in location 7FFFH (the highest address) of the FEEPROM, which should be programmed to 01010000B to activate the security feature.

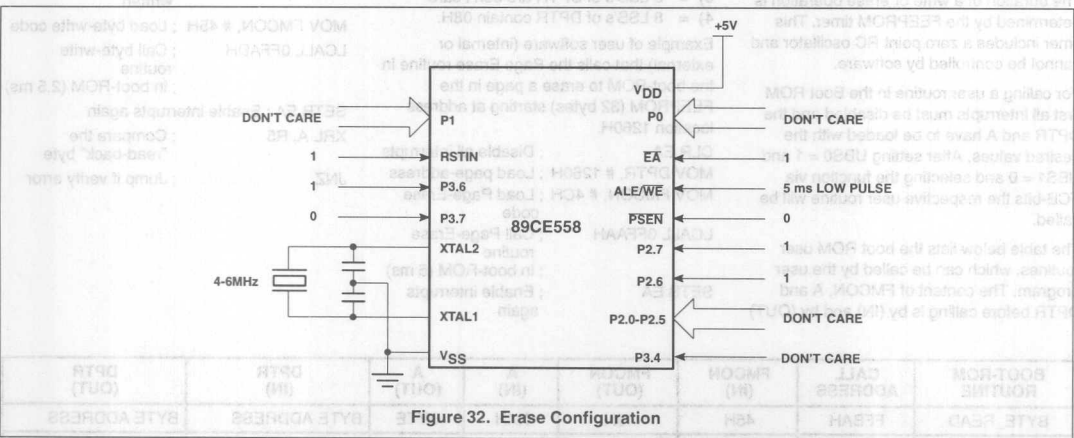
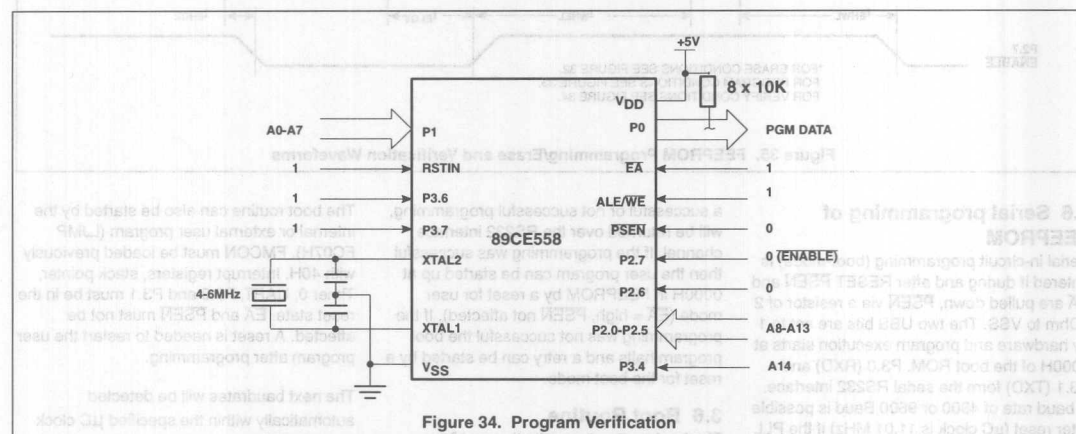
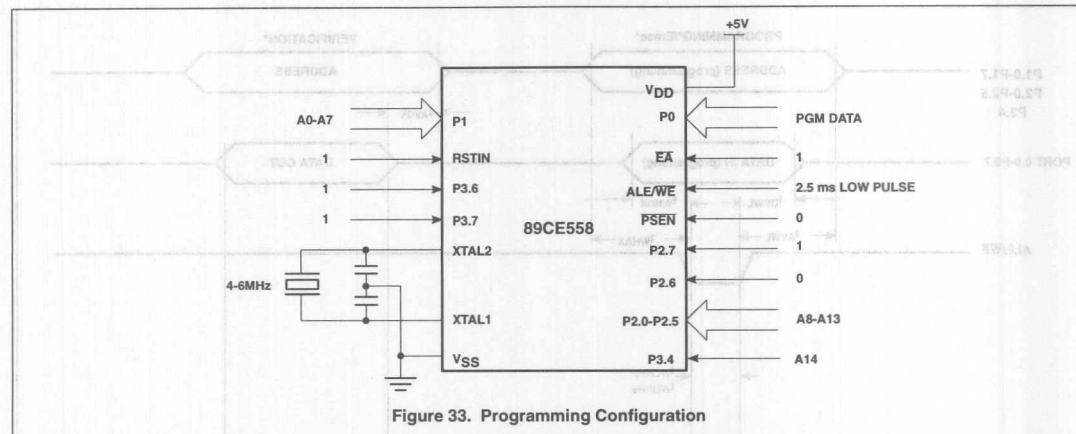


Figure 32. Erase Configuration

## Single-chip 8-bit microcontroller

80CE558/83CE558/89CE558



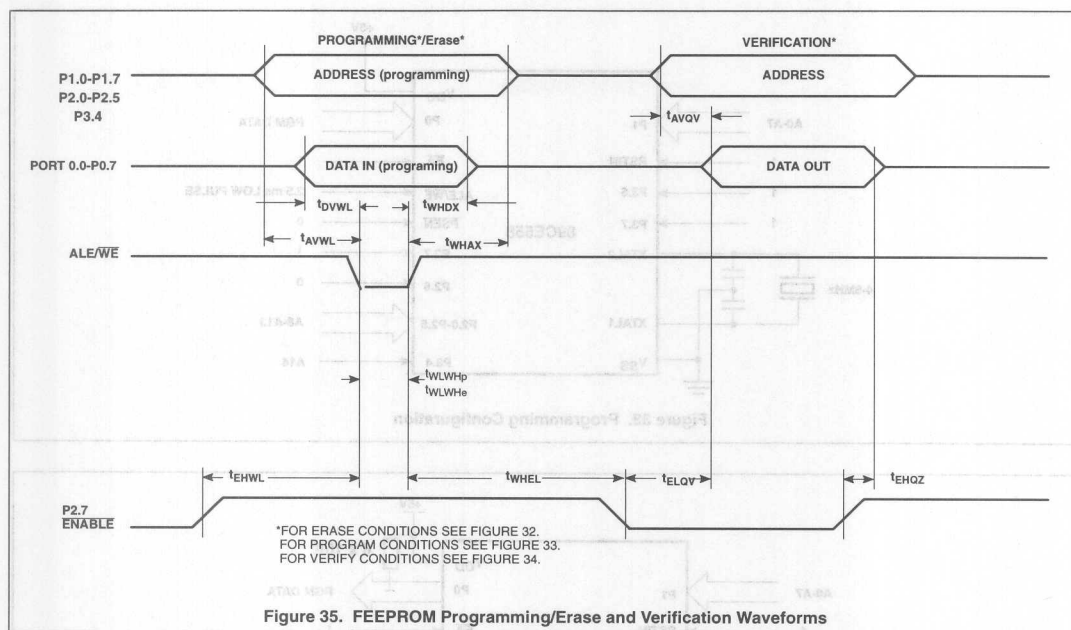
## EEPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

$T_{amb} = -40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $V_{DD} = 5V \pm 10\%$ ,  $V_{SS} = 0V$  (See Figure 35)

SYMBOL	PARAMETER	MIN	MAX	UNIT
$f_{CLCL}$	Oscillator frequency (standard oscillator)	4	6	MHz
$t_{AVWL}$	Address setup to WE LOW	$48t_{CLCL}$	-	
$t_{WHAX}$	Address hold after WE HIGH	$48t_{CLCL}$	-	
$t_{DVWL}$	Data setup to WE LOW	$48t_{CLCL}$	-	
$t_{WHDX}$	Data hold after WE HIGH	$48t_{CLCL}$	-	
$t_{EHWL}$	P2.7 (ENABLE) HIGH to WE LOW	$48t_{CLCL}$	-	
$t_{WHEL}$	WE HIGH to P2.7 (ENABLE) LOW	$48t_{CLCL}$	-	
$t_{WLWHp}$	WE width (programming)	2.25	2.75	ms
$t_{WLWHe}$	WE width (erase)	4.5	5.5	ms
$t_{AVQV}$	Address to data valid	-	$48t_{CLCL}$	
$t_{ELQV}$	P2.7 (ENABLE) Low to data valid	-	$48t_{CLCL}$	
$t_{EHQZ}$	Data float after P2.7 (ENABLE) HIGH	0	$48t_{CLCL}$	

## Single-chip 8-bit microcontroller

80CE558/83CE558/89CE558



### 3.6 Serial programming of FEEPROM

Serial in-circuit programming (boot-mode) is entered if during and after RESET  $\overline{PSEN}$  and  $\overline{EA}$  are pulled down,  $\overline{PSEN}$  via a resistor of 2 k $\Omega$  to VSS. The two UBS bits are set to 1 by hardware and program execution starts at 0000H of the boot ROM. P3.0 (RXD) and P3.1 (TXD) form the serial RS232 interface. A baud rate of 4800 or 9600 Baud is possible after reset ( $\mu$ C clock is 11.01 MHz) if the PLL oscillator is selected. The receive and transmit channel have the same baud rate. The boot routine inputs the Intel Hex Object Format. The baud rate will be selected automatically after reception of the first character (:) of the object file. No other characters are allowed to precede the first (:) character. Programming is only started if the first received record has the right type indication (TT). If the security feature is activated then the programming starts with a Full Erase, otherwise only the addressed page(s) will be erased and the not altered bytes are rewritten. During the erase or write operation the next string of bytes can be received. Xon and Xoff handshake codes are used to control the serial transfer. At the end of the programming a message that indicates

a successful or not successful programming, will be returned over the RS232 interface channel. If the programming was successful then the user program can be started up at 0000H in FEEPROM by a reset for user mode ( $\overline{EA}$  = high,  $\overline{PSEN}$  not affected). If the programming was not successful the boot program halts and a retry can be started by a reset for the boot mode.

#### 3.6 Boot Routine

The boot routine transmits the next "one ASCII character" messages via the RS232 interface:

- " . " After each record type TT = 00H indication in the HEX file.
- " X " Checksum error of a record in the HEX file detected.
- " Y " Wrong record type received
- " Z " Buffer overflow error (Check Xon/Xoff of terminal)
- " R " Verification error (of last written byte)
- " V " End record received and programming of FEEPROM was successful

No messages are transmitted if the baud rate of the first character (:) can not be detected.

The boot routine can also be started by the internal or external user program (LJMP FC07H). FMCON must be loaded previously with 40H. Interrupt registers, stack pointer, Timer 0, UART, P3.0 and P3.1 must be in the reset state.  $\overline{EA}$  and  $\overline{PSEN}$  must not be affected. A reset is needed to restart the user program after programming.

The next baudrates will be detected automatically within the specified  $\mu$ C clock range in MHz.

Baudrate	$f_{CLK}$ (min)	$f_{CLK}$ (max)
1200	1 <sup>1)</sup>	3.6
2400	2 <sup>1)</sup>	7.3
4800	4	19.7 <sup>1)</sup>
9600	7.9	29.5 <sup>1)</sup>
19200	15.7	59 <sup>1)</sup>

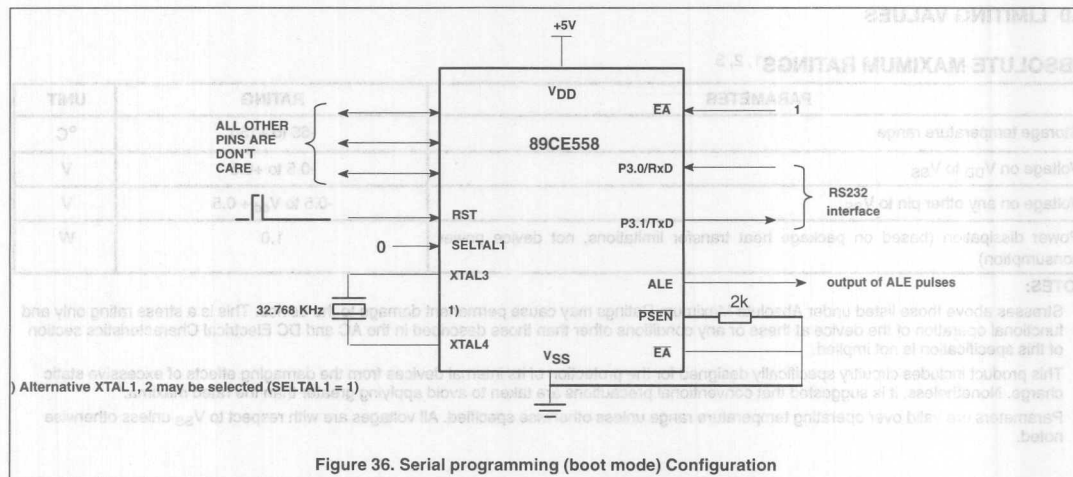
**Note 1):** value outside the specified clock range

Note that the boot routines can (re) program any number of bytes from 1 byte to 32k bytes, independent in which order or at which location.



## 8 Single-chip 8-bit microcontroller

80CE558/83CE558/89CE558



**NOTE:** In the following, each letter corresponds to one hexadecimal digit in ASCII representation (the digits 0 through 9, and the letters A through F).

## Definitions:

<b>:</b>	- Record start character	<b>AAAA</b>	- Load address in hexadecimal of first data byte in this record.	<b>CC</b>	- Record checksum. This is the 2's complement of the summation of all of the bytes in the record from the byte count through the last data byte. While the summation is calculated, it is always truncated to a one byte result. Thus, if all of the bytes in the record are summed, including the checksum itself, the result will always be 00 if the record is valid.
<b>BC</b>	- Byte Count. The hexadecimal number of data bytes in the record. This may theoretically be any number from 0 to 255 prefer to deal with 16 data bytes per record (as shown in the example below).	<b>TT</b>	- Record type. The record type is 00 for data records and 01 for the end record.		
		<b>HH</b>	- One hexadecimal data byte.		

Construction of data records (using the notation defined above) is as follows:

```
:: BCAAATTHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHCC
```

```
:: BCAAATTHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHCC
```

The last record in a file is the end record and contains no data. Usually the end record will appear as shown in the first example below.

However, in some cases a 16 bit checksum of all of the data bytes in the entire file may be inserted in the address field of the end record. This checksum would correspond to one generated by an EPROM programmer during file load, and its inclusion does not violate the rules for this format. This is shown in the second example.

```
:00000001FF
```

```
:00B12C0122
```

Successive hex records need not appear in sequential address order. For instance, a record for address 0000 might appear after a record for address 7FE0. All of the bytes in a single record, however, must be in sequence. Any characters that appear outside of a record (i.e. after a checksum, but before the next ":") should be ignored, if present.

An example of a valid hex file follows:

```
:10010000C2F0E53030E704F404D2F08531F030F786
```

```
:100110000763F0FF05F0B2F0A430F00A63F0FFF4DB
```

```
:0C0120002401500205F085F032F5332276
```

```
:00000001FF
```

8 Single-chip 8-bit microcontroller

80CE558/83CE558/89CE558

#### 4.0 LIMITING VALUES

**ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>**

PARAMETER	RATING	UNIT
Storage temperature range	-65 to +150	°C
Voltage on V <sub>DD</sub> to V <sub>SS</sub>	-0.5 to +6.5	V
Voltage on any other pin to V <sub>SS</sub>	-0.5 to V <sub>dd</sub> + 0.5	V
Power dissipation (based on package heat transfer limitations, not device power consumption)	1.0	W

**NOTES:**

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
2. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions are taken to avoid applying greater than the rated maxima.
3. Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to  $V_{SS}$  unless otherwise noted.

## Single-chip 8-bit microcontroller

## 80CE558/83CE558/89CE558

## 5.0 DC CHARACTERISTICS

## DC ELECTRICAL CHARACTERISTICS

$V_{DD} = 5V (\pm 10\%)$ ,  $V_{SS} = 0V$ ,  $T_{amb} = 0^{\circ}C$  to  $+70^{\circ}C$  (8xCE558EBx). All voltages with respect to  $V_{SS}$  unless otherwise specified.

SYMBOL	PARAMETER	TEST CONDITIONS	TEST CONDITIONS	LIMITS		UNIT
				MIN	MAX	
V <sub>DD</sub>	Supply voltage			4.5	5.5	V
I <sub>DD</sub>	Supply current operating : 89CE558 83CE558		See notes 1 and 2 f <sub>CLK</sub> = 16MHz V <sub>DD</sub> = 5.5 V		50 40	mA mA
I <sub>ID</sub>	Idle mode : 89CE558 83CE558		See notes 1 and 3 f <sub>CLK</sub> = 16MHz V <sub>DD</sub> = 5.5 V		10 8	mA mA
I <sub>PD</sub>	Power-down current		See note 4 2V < V <sub>PD</sub> < V <sub>DDmax</sub>		100	μA
Inputs						
V <sub>IL</sub>	Input LOW voltage, except E <sub>A</sub> , SCL, SDA			-0.5	0.2V <sub>DD</sub> -0.1	V
V <sub>IL1</sub>	Input LOW voltage to E <sub>A</sub>			-0.5	0.2V <sub>DD</sub> -0.3	V
V <sub>IL2</sub>	Input LOW voltage to SCL, SDA <sup>5</sup>			-0.5	0.3V <sub>DD</sub>	V
V <sub>IH</sub>	Input HIGH voltage, except XTAL1, RST, SCL, SDA			0.2V <sub>DD</sub> +0.9	V <sub>DD</sub> +0.5	V
V <sub>IH1</sub>	Input HIGH voltage, XTAL1, RST			0.7V <sub>DD</sub>	V <sub>DD</sub> +0.5	V
V <sub>IH2</sub>	Input HIGH voltage, SCL, SDA <sup>5</sup>			0.7V <sub>DD</sub>	6.0	V
I <sub>IL</sub>	Input current LOW level, Ports 0, 1, 2, 3, 4		V <sub>IN</sub> = 0.45V		-50	μA
I <sub>TL</sub>	Transition current HIGH to LOW, Ports 0, 1, 2, 3, 4		See note 6		-650	μA
±I <sub>IL1</sub>	Input leakage current, Port 0, E <sub>A</sub> , ADEXS, EW		0.45V < V <sub>I</sub> < V <sub>DD</sub>		10	μA
±I <sub>IL2</sub>	Input leakage current, SCL, SDA		0V < V <sub>I</sub> < 6V 0V < V <sub>DD</sub> < 5.5V		10	μA
±I <sub>IL3</sub>	Input leakage current, Port 5		0.45V < V <sub>I</sub> < V <sub>DD</sub>		1	μA
Outputs						
V <sub>OL</sub>	Output low voltage, Ports 1, 2, 3, 4		I <sub>OL</sub> = 1.6mA <sup>7,17</sup>		0.45	V
V <sub>OL1</sub>	Output low voltage, Port 0, ALE, PSEN, PWM0, PWM1		I <sub>OL</sub> = 3.2mA <sup>7,17</sup>		0.45	V
V <sub>OL2</sub>	Output low voltage, SCL, SDA		I <sub>OL</sub> = 3.0mA <sup>7,17</sup>		0.4	V
V <sub>OH</sub>	Output high voltage, Ports 1, 2, 3, 4		V <sub>DD</sub> =5V±10% -I <sub>OH</sub> = 60μA -I <sub>OH</sub> = 25μA -I <sub>OH</sub> = 10μA	2.4 0.75V <sub>DD</sub> 0.9V <sub>DD</sub>	V V V	
V <sub>OH1</sub>	Output high voltage (Port 0 in external bus mode, ALE, PSEN, RST, PWM0, PWM1) <sup>8</sup>		V <sub>DD</sub> =5V±10% -I <sub>OH</sub> = 800μA -I <sub>OH</sub> = 300μA -I <sub>OH</sub> = 80μA	2.4 0.75V <sub>DD</sub> 0.9V <sub>DD</sub>	V V V	
R <sub>RST</sub>	Internal reset pull-down resistor			50	150	kΩ
C <sub>IO</sub>	Pin capacitance		Test freq = 1MHz, T <sub>amb</sub> = 25°C		10	pF

Single-chip 8-bit microcontroller

80CE558/83CE558/89CE558

DC ELECTRICAL CHARACTERISTICS (Continued)

$V_{DD} = 5V (\pm 10\%)$ ,  $V_{SS} = 0V$ ,  $T_{amb} = -40^{\circ}C$  to  $+85^{\circ}C$  (8xCE558EFx). DC parameters not included here are the same as in the 8xCE558EBB.

DC electrical characteristics

All voltages with respect to  $V_{SS}$  unless otherwise specified.

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			MIN	MAX	
Inputs					
V <sub>IL</sub>	Input LOW voltage, except EA, SCL, SDA		-0.5	0.2V <sub>DD</sub> -0.15	V
V <sub>IL1</sub>	Input LOW voltage to EA		-0.5	0.2V <sub>DD</sub> -0.35	V
V <sub>IH</sub>	Input HIGH voltage, except XTAL1, RST, SCL, SDA		0.2V <sub>DD</sub> +1.0	V <sub>DD</sub> +0.5	V
V <sub>IH1</sub>	Input HIGH voltage, XTAL1, RST		0.7V <sub>DD</sub> +0.1	V <sub>DD</sub> +0.5	V
I <sub>IL</sub>	Input current LOW level, Ports 1, 2, 3, 4	V <sub>IN</sub> = 0.45V		-75	μA
I <sub>TL</sub>	Transition current HIGH to LOW, Ports 1, 2, 3, 4	See note 6		-750	μA

DC ELECTRICAL CHARACTERISTICS ANALOG

$AV_{DD} = 5V (\pm 10\%)$ ,  $AV_{SS} = 0V$ ,  $T_{amb} = 0^{\circ}C$  to  $+70^{\circ}C$  (8xCE558EBx).

$AV_{DD} = 5V (\pm 10\%)$ ,  $AV_{SS} = 0V$ ,  $T_{amb} = -40^{\circ}C$  to  $+85^{\circ}C$  (8xCE558EFx).

All voltages with respect to  $V_{SS}$  unless otherwise specified.

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			MIN	MAX	
$AV_{DD}$	Analog supply voltage	$AV_{DD} = V_{DD} \pm 0.2V$	4.5	5.5	V
$AI_{DD}$	Analog supply current	Port 5 = 0 to $AV_{DD}$ see note 19		1.2	mA
$AI_{ID}$	Idle mode	see note 18		50	$\mu A$
$AI_{PD}$	Power-down mode	$2V < AV_{PD} < AV_{DD}$ see note 18		50	$\mu A$
<b>Analog Inputs</b>					
$AV_{IN}$	Analog input voltage		$AV_{SS}-0.2$	$AV_{DD}+0.2$	V
$AV_{REF}$	Reference voltage: AV <sub>REF-</sub> AV <sub>REF+</sub>		$AV_{SS}-0.2$	$AV_{DD}+0.2$	V
$R_{REF}$	Resistance between AV <sub>REF+</sub> and AV <sub>REF-</sub>		10	50	k $\Omega$
$C_{IA}$	Analog input capacitance			15	pF
$DL_e$	Differential non-linearity <sup>9, 10, 11</sup>			$\pm 1$	LSB
$IL_e$	Integral non-linearity <sup>9, 12</sup>			$\pm 2$	LSB
$OS_e$	Offset error <sup>9, 13</sup>			$\pm 2$	LSB
$G_e$	Gain error <sup>9, 14</sup>			$\pm 0.4$	%
$A_e$	Absolute voltage error <sup>9, 15</sup>			$\pm 3$	LSB
$M_{CTC}$	Channel to channel matching			$\pm 1$	LSB
$C_t$	Crosstalk between inputs of port 5 <sup>16</sup>	0-100kHz		-60	dB

## Single-chip 8-bit microcontroller

80CE558/83CE558/89CE558

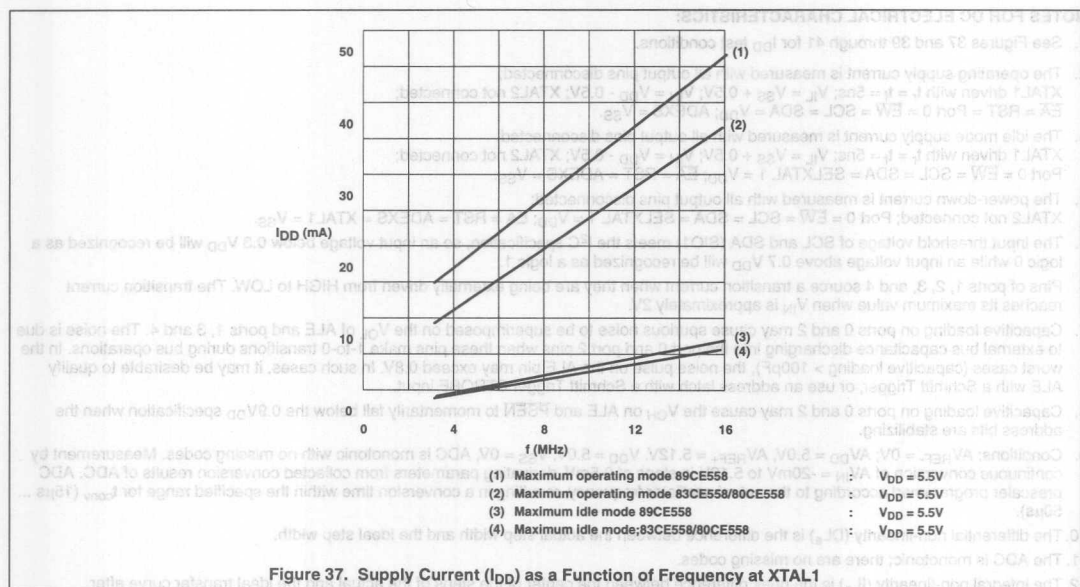
## NOTES FOR DC ELECTRICAL CHARACTERISTICS:

- See Figures 37 and 39 through 41 for  $I_{DD}$  test conditions.
- The operating supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 5\text{ns}$ ;  $V_{IL} = V_{SS} + 0.5\text{V}$ ;  $V_{IH} = V_{DD} - 0.5\text{V}$ ; XTAL2 not connected;  $\overline{\text{EA}} = \text{RST} = \text{Port 0} = \overline{\text{EW}} = \text{SCL} = \text{SDA} = V_{DD}$ ;  $\text{ADEXS} = V_{SS}$ .
- The idle mode supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 5\text{ns}$ ;  $V_{IL} = V_{SS} + 0.5\text{V}$ ;  $V_{IH} = V_{DD} - 0.5\text{V}$ ; XTAL2 not connected; Port 0 =  $\overline{\text{EW}} = \text{SCL} = \text{SDA} = \text{SELXTAL 1} = V_{DD}$ ;  $\overline{\text{EA}} = \text{RST} = \text{ADEXS} = V_{SS}$ .
- The power-down current is measured with all output pins disconnected; XTAL2 not connected; Port 0 =  $\overline{\text{EW}} = \text{SCL} = \text{SDA} = \text{SELXTAL 1} = V_{DD}$ ;  $\overline{\text{EA}} = \text{RST} = \text{ADEXS} = \text{XTAL1} = V_{SS}$ .
- The input threshold voltage of SCL and SDA (SIO1) meets the I<sup>2</sup>C specification, so an input voltage below  $0.3 V_{DD}$  will be recognized as a logic 0 while an input voltage above  $0.7 V_{DD}$  will be recognized as a logic 1.
- Pins of ports 1, 2, 3, and 4 source a transition current when they are being externally driven from HIGH to LOW. The transition current reaches its maximum value when  $V_{IN}$  is approximately 2V.
- Capacitive loading on ports 0 and 2 may cause spurious noise to be superimposed on the  $V_{OL}$  of ALE and ports 1, 3 and 4. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading  $> 100\text{pF}$ ), the noise pulse on the ALE pin may exceed 0.8V. In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.
- Capacitive loading on ports 0 and 2 may cause the  $V_{OH}$  on ALE and PSEN to momentarily fall below the  $0.9V_{DD}$  specification when the address bits are stabilizing.
- Conditions:  $\text{AV}_{\text{REF-}} = 0\text{V}$ ;  $\text{AV}_{\text{DD}} = 5.0\text{V}$ ,  $\text{AV}_{\text{REF+}} = 5.12\text{V}$ .  $V_{DD} = 5.0\text{V}$ ,  $V_{SS} = 0\text{V}$ , ADC is monotonic with no missing codes. Measurement by continuous conversion of  $\text{AV}_{\text{IN}} = -20\text{mV}$  to  $5.12\text{V}$  in steps of  $0.5\text{mV}$ , deriving parameters from collected conversion results of ADC. ADC prescaler programmed according to the actual oscillator frequency, resulting in a conversion time within the specified range for  $t_{\text{conv}}$  ( $15\mu\text{s}$  ...  $50\mu\text{s}$ ).
- The differential non-linearity ( $\text{DL}_e$ ) is the difference between the actual step width and the ideal step width.
- The ADC is monotonic; there are no missing codes.
- The integral non-linearity ( $\text{IL}_e$ ) is the peak difference between the center of the steps of the actual and the ideal transfer curve after appropriate adjustment of gain and offset error.
- The offset error ( $\text{OS}_e$ ) is the absolute difference between the straight line which fits the actual transfer curve (after removing gain error), and a straight line which fits the ideal transfer curve. The offset error is constant at every point of the actual transfer curve.
- The gain error ( $\text{G}_e$ ) is the relative difference in percent between the straight line fitting the actual transfer curve (after removing offset error), and the straight line which fits the ideal transfer curve. Gain error is constant at every point on the transfer curve.
- The absolute voltage error ( $\text{A}_e$ ) is the maximum difference between the center of the steps of the actual transfer curve of the non-calibrated ADC and the ideal transfer curve.
- This should be considered when both analog and digital signals are simultaneously input to port 5.
- Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:
  - Maximum  $I_{OL}$  per port pin: 10 mA
  - Maximum  $I_{OL}$  8-bit port -
    - Port 0 : 26 mA
    - Ports 1, 2, 3 and 4 : 15 mA
  - Maximum total  $I_{OL}$  for all output pins : 86 mA
- If  $I_{OL}$  exceeds the test conditions,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
- $\text{AI}_{\text{ID}}$  is measured with  $\text{SELXTAL1} = V_{DD}$ .
- $\text{AI}_{\text{ID}}$  and  $\text{AI}_{\text{PD}}$  are measured with the 32 kHz oscillator being halted, XTAL3 not connected and  $\text{XTAL4} = V_{SS}$ .



## Single-chip 8-bit microcontroller

## 80CE558/83CE558/89CE558



1. The internal non-linearity ( $\Delta_1$ ) is the absolute difference between the ideal transfer curve and the actual transfer curve after appropriate adjustment of gain and offset error.

12. The offset error ( $\Delta_2$ ) is the absolute difference between the straight line which fits the actual transfer curve (after removing gain error) and a straight line which fits the ideal transfer curve. The offset error is constant at every point of the actual transfer curve.

14. The gain error ( $\Delta_3$ ) is the relative difference in percent between the straight line fitting the actual transfer curve (after removing offset error) and the straight line which fits the ideal transfer curve. Gain error is constant at every point on the transfer curve.

15. The absolute voltage error ( $\Delta_4$ ) is the maximum difference between the center of the steps of the actual transfer curve of the non-calibrated ADC and the ideal transfer curve.

16. This should be considered when both analog and digital signals are simultaneously input to port 2.

17. Under steady state (non-transient) conditions,  $I_{DD}$  must be externally limited as follows:

- Maximum  $I_{DD}$  per port pin: 10 mA
- Maximum  $I_{DD}$  8-bit port: 88 mA
- Port 0: 28 mA
- Ports 1, 2, 3 and 4: 12 mA
- Maximum total  $I_{DD}$  for all output pins: 88 mA

If  $I_{DD}$  exceeds the last conditions,  $V_{DD}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

18.  $I_{DD}$  is measured with SELXTAL1 =  $V_{DD}$ .

19.  $I_{DD}$  and  $A_{DCL}$  are measured with the SS bit of capacitor being pulled, XTAL2, XTAL3 not connected and XTAL4 =  $V_{DD}$ .

## Single-chip 8-bit microcontroller

80CE558/83CE558/89CE558

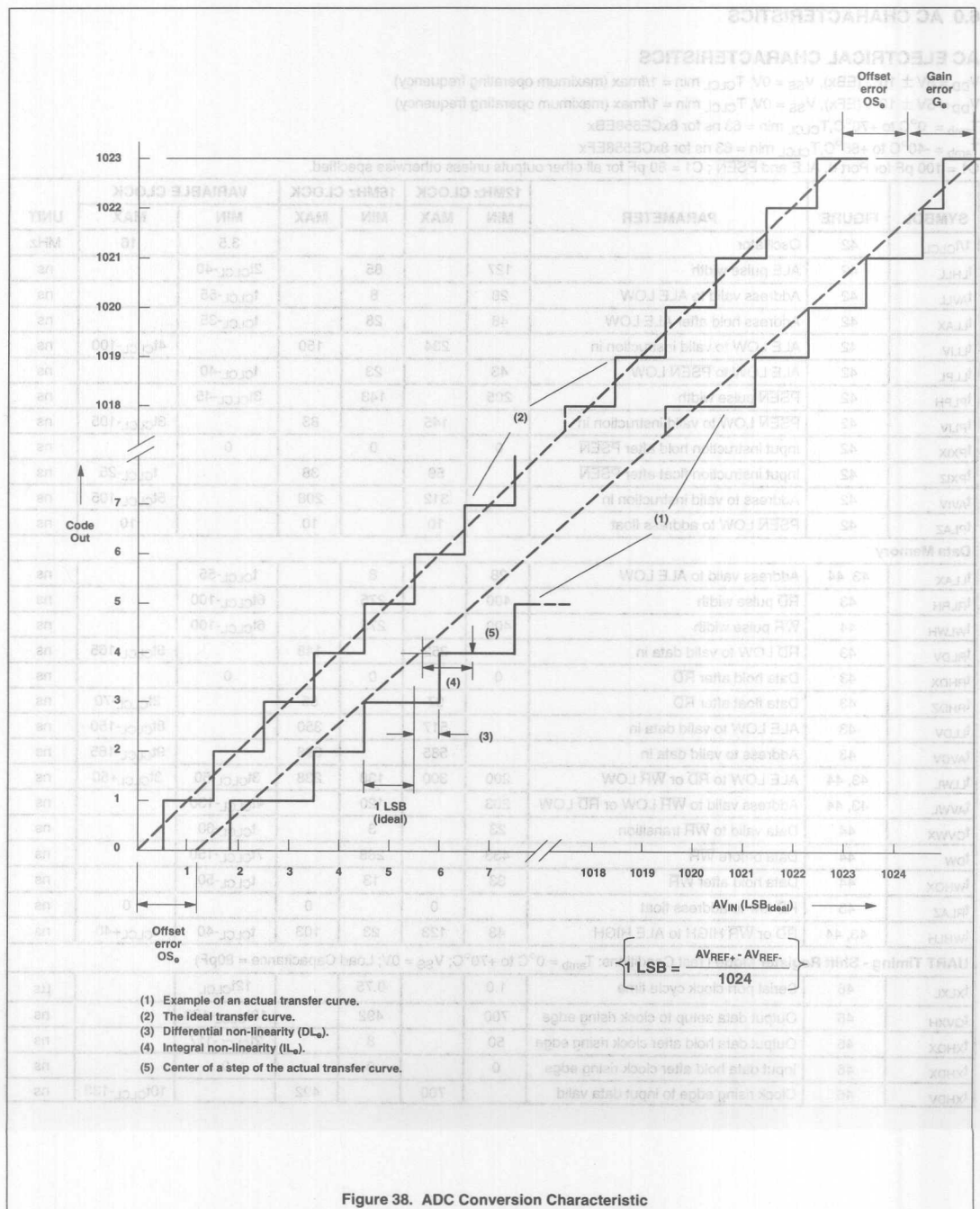


Figure 38. ADC Conversion Characteristic

## 8 Single-chip 8-bit microcontroller

80CE558/83CE558/89CE558

## 6.0 AC CHARACTERISTICS

## AC ELECTRICAL CHARACTERISTICS

$V_{DD} = 5V \pm 10\%$  (EBx),  $V_{SS} = 0V$ ,  $T_{CLCL} \text{ min} = 1/f_{\text{max}}$  (maximum operating frequency)

$V_{DD} = 5V \pm 10\%$  (EFx),  $V_{SS} = 0V$ ,  $T_{CLCL} \text{ min} = 1/f_{\text{max}}$  (maximum operating frequency)

$T_{\text{amb}} = 0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$ ,  $T_{CLCL} \text{ min} = 63 \text{ ns}$  for 8xCE558EBx

$T_{\text{amb}} = -40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $T_{CLCL} \text{ min} = 63 \text{ ns}$  for 8xCE558EFx

$C_1 = 100 \text{ pF}$  for Port 0, ALE and PSEN;  $C_1 = 80 \text{ pF}$  for all other outputs unless otherwise specified.

SYMBOL	FIGURE	PARAMETER	12MHz CLOCK		16MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	
$1/t_{CLCL}$	42	Oscillator					3.5	16	MHz
$t_{LHLL}$	42	ALE pulse width	127		85		$2t_{CLCL}-40$		ns
$t_{AVLL}$	42	Address valid to ALE LOW	28		8		$t_{CLCL}-55$		ns
$t_{LLAX}$	42	Address hold after ALE LOW	48		28		$t_{CLCL}-35$		ns
$t_{LLIV}$	42	ALE LOW to valid instruction in		234		150		$4t_{CLCL}-100$	ns
$t_{LLPL}$	42	ALE LOW to PSEN LOW	43		23		$t_{CLCL}-40$		ns
$t_{PLPH}$	42	PSEN pulse width	205		143		$3t_{CLCL}-45$		ns
$t_{PLIV}$	42	PSEN LOW to valid instruction in		145		83		$3t_{CLCL}-105$	ns
$t_{PXIX}$	42	Input instruction hold after PSEN	0		0		0		ns
$t_{PXIZ}$	42	Input instruction float after PSEN		59		38		$t_{CLCL}-25$	ns
$t_{AVIV}$	42	Address to valid instruction in		312		208		$5t_{CLCL}-105$	ns
$t_{PLAZ}$	42	PSEN LOW to address float		10		10		10	ns
<b>Data Memory</b>									
$t_{LLAX}$	43, 44	Address valid to ALE LOW	28		8		$t_{CLCL}-55$		ns
$t_{RLRH}$	43	RD pulse width	400		275		$6t_{CLCL}-100$		ns
$t_{WLWH}$	44	WR pulse width	400		275		$6t_{CLCL}-100$		ns
$t_{RLDV}$	43	RD LOW to valid data in		252		148		$5t_{CLCL}-165$	ns
$t_{RHDX}$	43	Data hold after RD	0		0		0		ns
$t_{RHDZ}$	43	Data float after RD		97		55		$2t_{CLCL}-70$	ns
$t_{LLDV}$	43	ALE LOW to valid data in		517		350		$8t_{CLCL}-150$	ns
$t_{AVDV}$	43	Address to valid data in		585		398		$9t_{CLCL}-165$	ns
$t_{LLWL}$	43, 44	ALE LOW to RD or WR LOW	200	300	138	238	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
$t_{AVWL}$	43, 44	Address valid to WR LOW or RD LOW	203		120		$4t_{CLCL}-130$		ns
$t_{QVWX}$	44	Data valid to WR transition	23		3		$t_{CLCL}-60$		ns
$t_{DW}$	44	Data before WR	433		288		$7t_{CLCL}-150$		ns
$t_{WHQX}$	44	Data hold after WR	33		13		$t_{CLCL}-50$		ns
$t_{RLAZ}$	43	RD low to address float		0		0		0	ns
$t_{WHLH}$	43, 44	RD or WR HIGH to ALE HIGH	43	123	23	103	$t_{CLCL}-40$	$t_{CLCL}+40$	ns
<b>UART Timing - Shift Register Mode</b> (Test Conditions: $T_{\text{amb}} = 0^{\circ}\text{C}$ to $+70^{\circ}\text{C}$ ; $V_{SS} = 0V$ ; Load Capacitance = 80pF)									
$t_{XLXL}$	46	Serial port clock cycle time	1.0		0.75		$12t_{CLCL}$		$\mu\text{s}$
$t_{QVXH}$	46	Output data setup to clock rising edge	700		492		$10t_{CLCL}-133$		ns
$t_{XHQX}$	46	Output data hold after clock rising edge	50		8		$2t_{CLCL}-117$		ns
$t_{XHDX}$	46	Input data hold after clock rising edge	0		0		0		ns
$t_{XHDV}$	46	Clock rising edge to input data valid		700		492		$10t_{CLCL}-133$	ns

Figure 38. AC Conversion Characteristics

Single-chip 8-bit microcontroller

80CE558/83CE558/89CE558

AC ELECTRICAL CHARACTERISTICS (Continued)

SYMBOL	PARAMETER	Standard-mode I <sup>2</sup> C-bus		UNIT
		MIN	MAX	
I <sup>2</sup> C Interface timing (refer to Figure 45)				
f <sub>SCL</sub>	SCL clock frequency	0	100	kHz
t <sub>BUF</sub>	Bus free time between a STOP and START condition	4.7	-	μs
t <sub>HD; STA</sub>	Hold time (repeated) START condition. After this period, the first clock pulse is generated	4.0	-	μs
t <sub>LOW</sub>	LOW period of the SCL clock	4.7	-	μs
t <sub>HIGH</sub>	High period of the SCL clock	4.0	-	μs
t <sub>SU; STA</sub>	Set-up time for a repeated START condition	4.7	-	μs
t <sub>HD; DAT</sub>	Data hold time	0 <sup>1</sup>	-	μs
t <sub>HD; DAT1,2,3</sub>	Data set-up time	250	-	ns
t <sub>RD, t<sub>RC</sub></sub>	Rise time of both SDA and SCL signals	-	1000	ns
t <sub>FD, t<sub>FC</sub></sub>	Fall time of both SDA and SCL signals	-	300	ns
t <sub>SU; STO</sub>	Set-up time for STOP condition	4.0	-	μs
C <sub>b</sub>	Capacitive load for each bus line	-	400	pF

All values referred to V<sub>IH</sub> and V<sub>IL max</sub> levels.

NOTES:

1. A device must internally provide a hold time of at least 300 ns from the SDA signal (referred to the V<sub>IH min</sub> of the SCL signal) in order to bridge the undefined region of the falling edge of SCL.

For 62ns < t<sub>CLCL</sub> < 570ns, 16MHz > f<sub>CLK</sub> > 3.5MHz) the Si01 interface meets concerning AC parameters the I<sup>2</sup>C-bus specification for bit-rates up to 100 kbit/s.

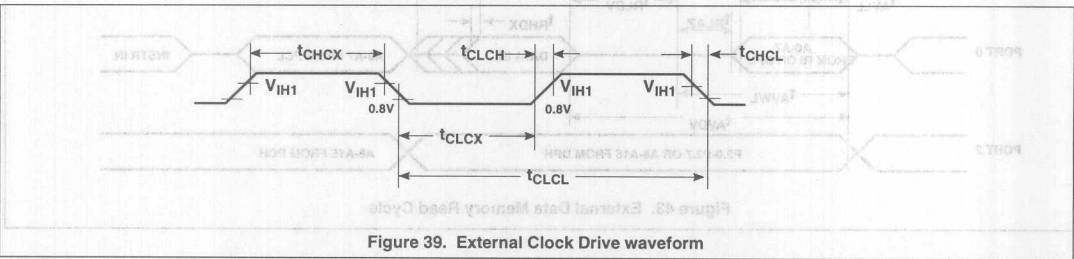
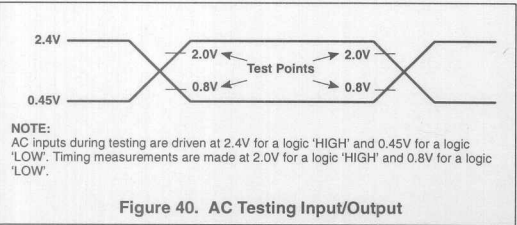
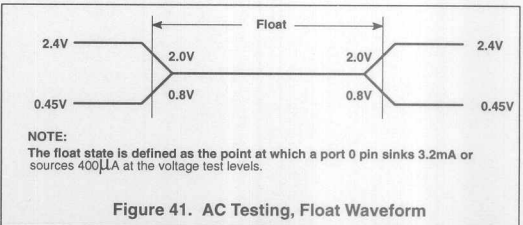


Figure 39. External Clock Drive waveform



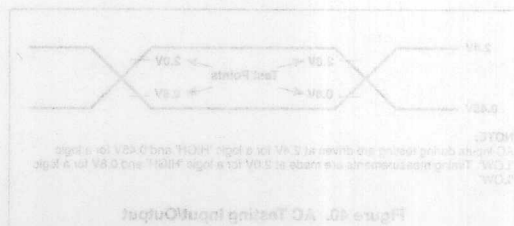
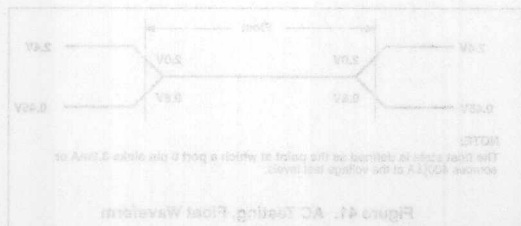
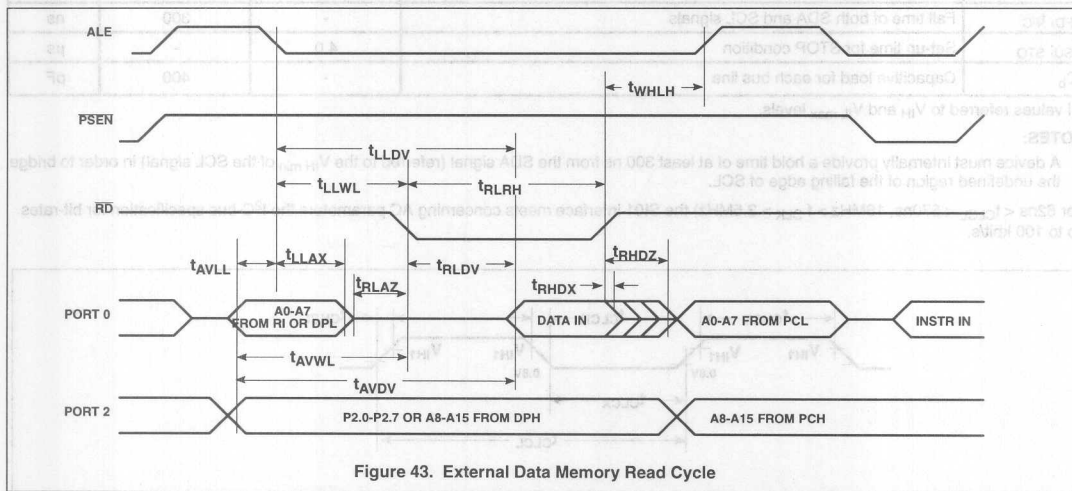
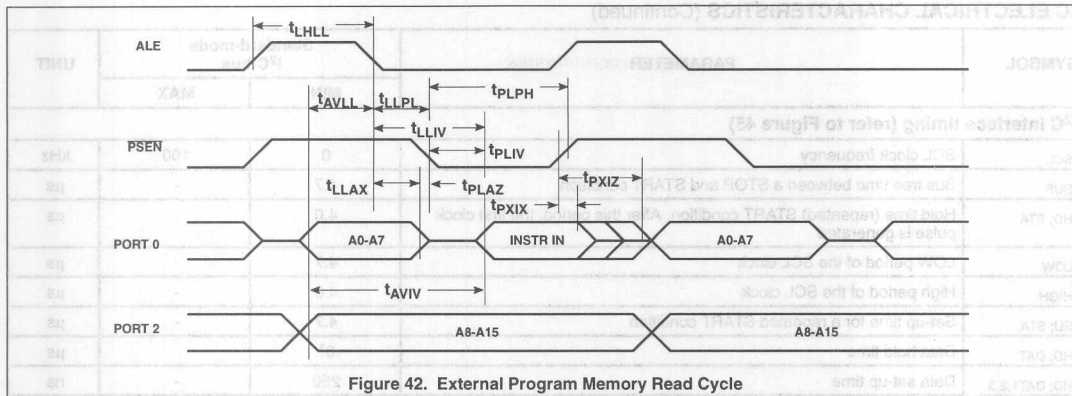
NOTE:  
AC inputs during testing are driven at 2.4V for a logic 'HIGH' and 0.45V for a logic 'LOW'. Timing measurements are made at 2.0V for a logic 'HIGH' and 0.8V for a logic 'LOW'.

Figure 40. AC Testing Input/Output



NOTE:  
The float state is defined as the point at which a port 0 pin sinks 3.2mA or sources 400μA at the voltage test levels.

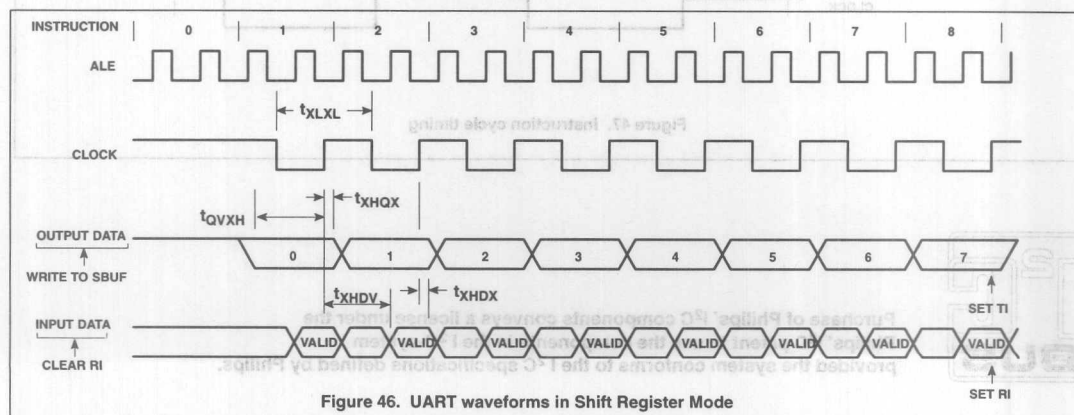
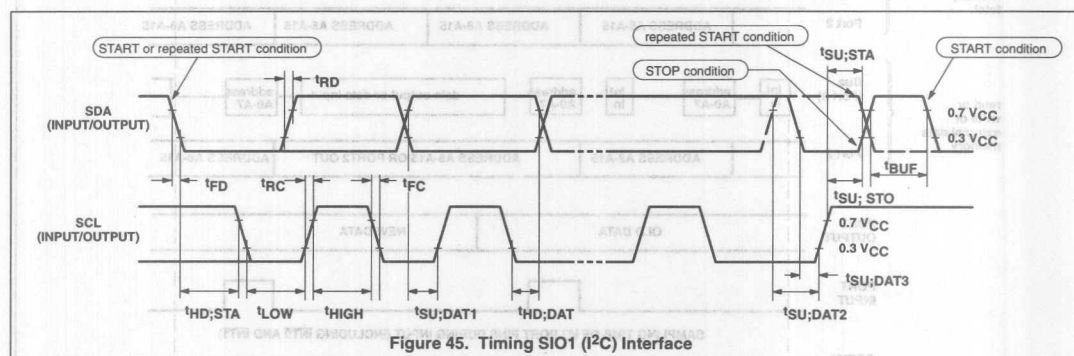
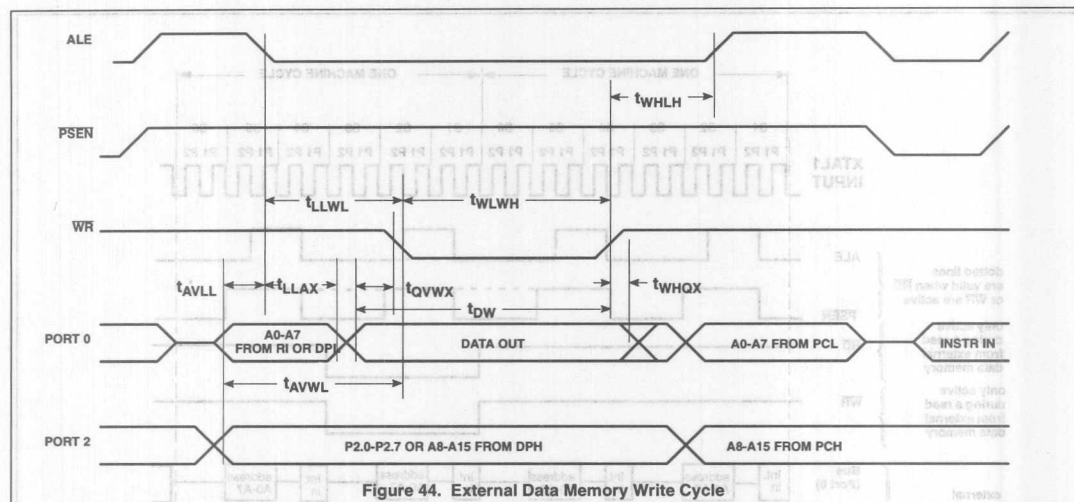
Figure 41. AC Testing, Float Waveform





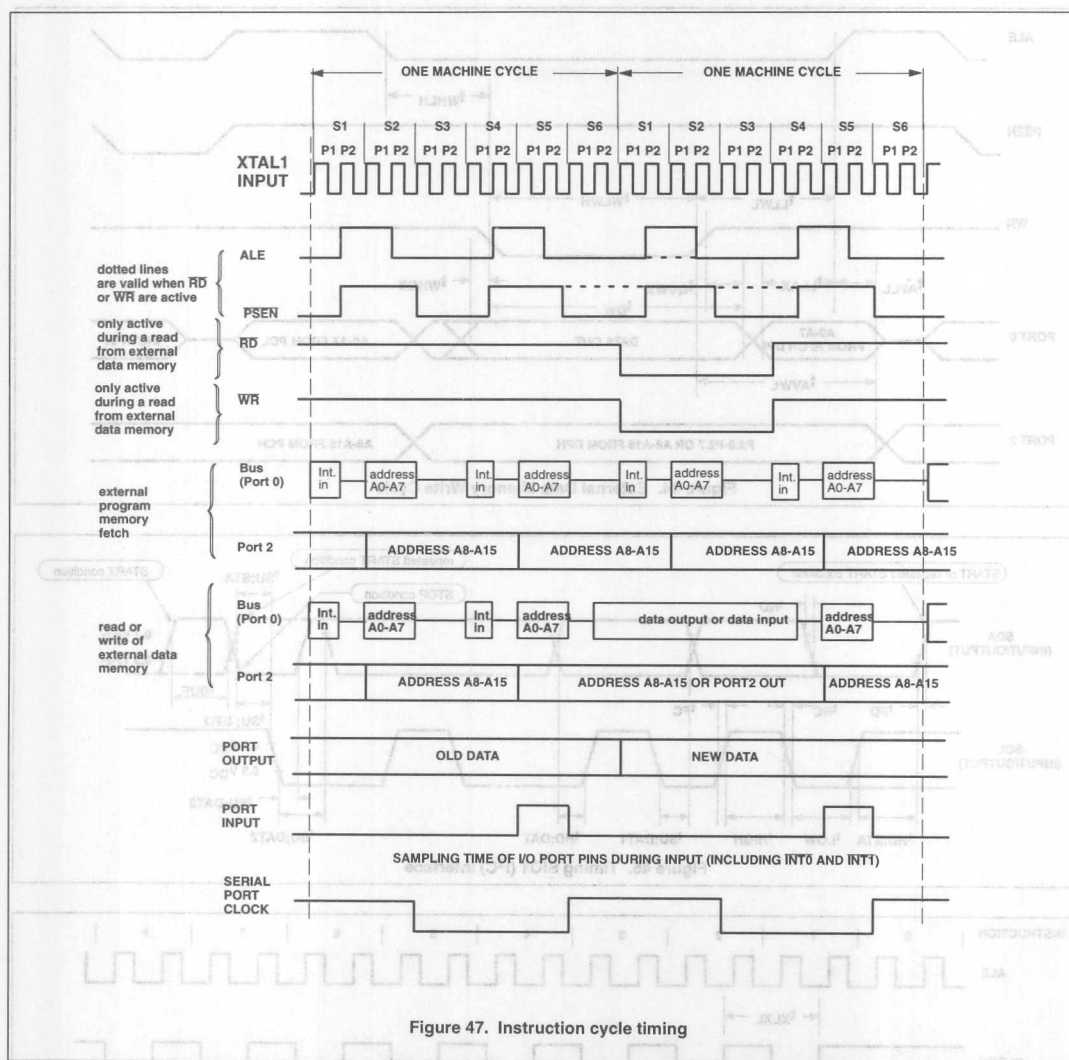
## Single-chip 8-bit microcontroller

80CE558/83CE558/89CE558



## Single-chip 8-bit microcontroller

80CE558/83CE558/89CE558



Purchase of Philips' I<sup>2</sup>C components conveys a license under the Philips' I<sup>2</sup>C patent to use the components in the I<sup>2</sup>C-system provided the system conforms to the I<sup>2</sup>C specifications defined by Philips.

## Single-chip 8-bit microcontroller

## 80C562/83C562

Single-chip 8-bit microcontroller with 8-bit A/D, capture/compare timer, high-speed outputs, PWM

## DESCRIPTION

The 80C562/83C562 (hereafter generically referred to as 8XC562) Single-Chip 8-Bit Microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 83C562/83C562 has the same instruction set as the 80C51.

The 8XC562 contains a non-volatile  $256 \times 8$  read-only program memory, a volatile  $256 \times 8$  read/write data memory (83C562) (the 80C562 is ROMless), a volatile  $256 \times 8$  read/write data memory, six 8-bit I/O ports, two 16-bit timer/event counters (identical to the timers of the 80C51), an additional 16-bit timer coupled to capture and compare latches, a 15-source, two-priority-level, nested interrupt structure, an 8-input ADC, two pulse width modulated outputs, standard 80C51 UART, a "watchdog" timer and on-chip oscillator and timing circuits. For systems that require extra capability, the 83C562 can be expanded using standard TTL compatible memories and logic.

The device also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of over 100 instructions: 49 one-byte, 45 two-byte and 17 three-byte. With a 12MHz crystal, 58% of the instructions are executed in  $1\mu\text{s}$  and 40% in  $2\mu\text{s}$ . Multiply and divide instructions require  $4\mu\text{s}$ .

## FEATURES

- 80C51 instruction set
- $8k \times 8$  ROM expandable externally to 64k bytes
- $256 \times 8$  RAM, expandable externally to 64k bytes
- Two standard 16-bit timer/counters
- An additional 16-bit timer/counter coupled to four capture registers and three compare registers
- Capable of producing eight synchronized, timed outputs
- An 8-bit ADC with eight multiplexed analog inputs
- Two 8-bit resolution, pulse width modulated outputs
- Five 8-bit I/O ports plus one 8-bit input port shared with analog inputs
- Full-duplex UART compatible with the standard 80C51
- On-chip watchdog timer
- Three temperature ranges
  - 0 to  $+70^\circ\text{C}$
  - $-40$  to  $+85^\circ\text{C}$
  - $-40$  to  $+125^\circ\text{C}$

## PIN CONFIGURATION

Pin	Function	Pin	Function
1	P5.0/ADC0	35	XTAL1
2	V <sub>DD</sub>	36	V <sub>SS</sub>
3	STADC	37	V <sub>SS</sub>
4	PWM0	38	NC
5	PWMT	39	P2.0/A08
6	EW	40	P2.1/A09
7	P4.0/CMSR0	41	P2.2/A10
8	P4.1/CMSR1	42	P2.3/A11
9	P4.2/CMSR2	43	P2.4/A12
10	P4.3/CMSR3	44	P2.5/A13
11	P4.4/CMSR4	45	P2.6/A14
12	P4.5/CMSR5	46	P2.7/A15
13	P4.6/CMT0	47	PSEN
14	P4.7/CMT1	48	ALE
15	RST	49	EA
16	P1.0/CT01	50	P0.7/AD7
17	P1.1/CT11	51	P0.6/AD6
18	P1.2/CT21	52	P0.5/AD5
19	P1.3/CT31	53	P0.4/AD4
20	P1.4/T2	54	P0.3/AD3
21	P1.5/RT2	55	P0.2/AD2
22	P1.6	56	P0.1/AD1
23	P1.7	57	P0.0/AD0
24	P3.0/RxD	58	AVref-
25	P3.1/TxD	59	AVref+
26	P3.2/INT0	60	AV <sub>SS</sub>
27	P3.3/INT1	61	AV <sub>DD</sub>
28	P3.4/T0	62	P5.7/ADC7
29	P3.5/T1	63	P5.6/ADC6
30	P3.6/WR	64	P5.5/ADC5
31	P3.7/RD	65	P5.4/ADC4
32	NC	66	P5.3/ADC3
33	NC	67	P5.2/ADC2
34	XTAL2	68	P5.1/ADC1

## Single-chip 8-bit microcontroller

## 80C562/83C562

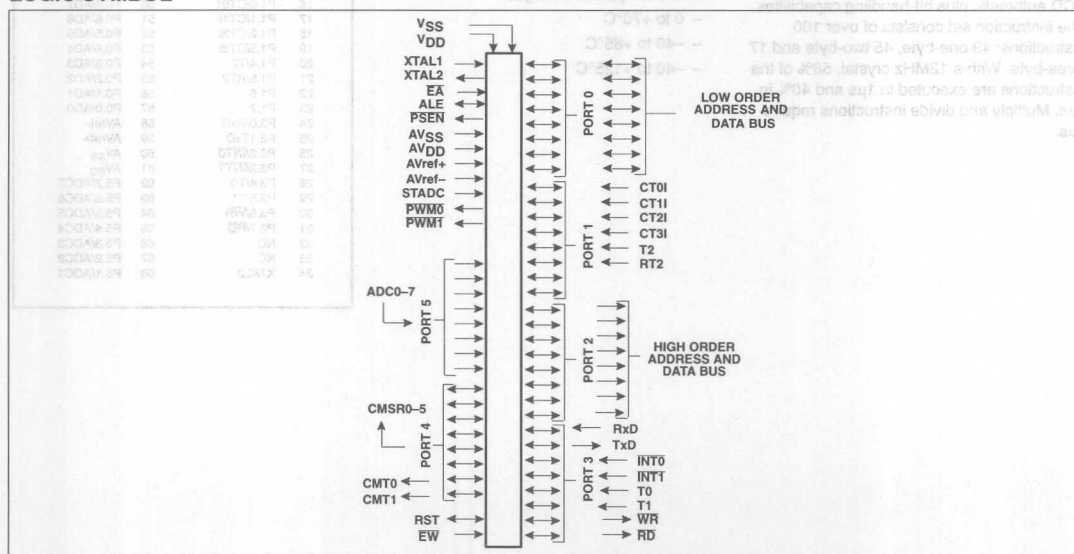
## ORDERING INFORMATION

PHILIPS PART ORDER NUMBER PART MARKING		PHILIPS NORTH AMERICA PART ORDER NUMBER		Drawing Number	EPROM	Drawing Number	TEMPERATURE RANGE °C	FREQ MHz
ROMless	ROM	ROMless	ROM				AND PACKAGE	
PCB80C562-16WP	PCB83C562-16WP/xxx	S80C562-4A68	S83C562-4A68	SOT188	S87C552-4A68 <sup>2</sup>	0398E	0 to +70, Plastic Leaded Chip Carrier	16
					S87C552-4K68 <sup>2</sup>	1473A	0 to +70, Plastic Leaded Chip Carrier w/Window	16
PCF80C562-12WP	PCF83C562-12WP/xxx	S80C562-2A68	S83C562-2A68	SOT188	S87C552-5A68 <sup>2</sup>	0398E	-40 to +85, Plastic Leaded Chip Carrier	12
					S87C552-5K68 <sup>2</sup>	1473A	-40 to +85, Plastic Leaded Chip Carrier w/Window	12
PCA80C562-12WP	PCA83C562-12WP/xxx	S80C562-6A68	S83C562-6A68	SOT188			-40 to +125, Plastic Leaded Chip Carrier	12

## NOTES:

- 80C562 and 83C562 frequency range is 1.2MHz–12MHz or 1.2MHz–16MHz.
- 87C552 frequency range is 3.5MHz–16MHz. For full specification, see the 87C552 data sheets.
- xxx denotes the ROM code number.

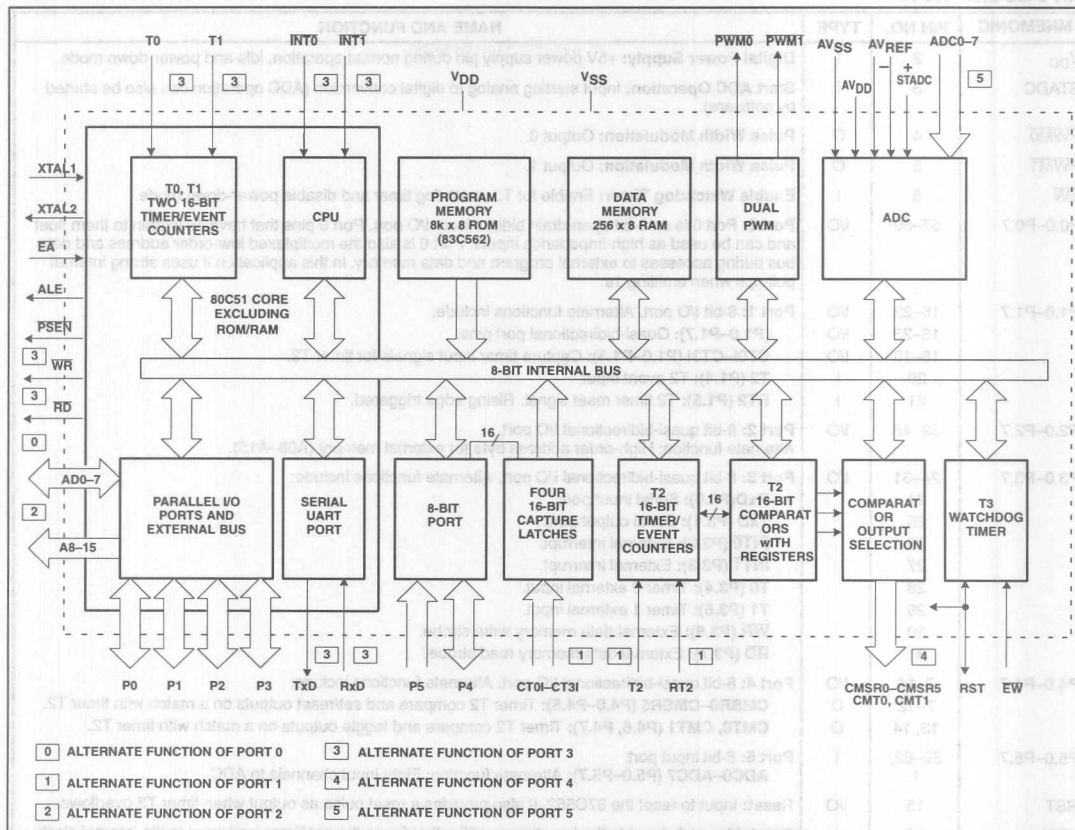
## LOGIC SYMBOL



## Single-chip 8-bit microcontroller

80C562/83C562

## BLOCK DIAGRAM





## Single-chip 8-bit microcontroller

80C562/83C562

## PIN DESCRIPTION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
V <sub>DD</sub>	2	I	<b>Digital Power Supply:</b> +5V power supply pin during normal operation, idle and power-down mode.
STADC	3	I	<b>Start ADC Operation:</b> Input starting analog to digital conversion (ADC operation can also be started by software).
PWM0	4	O	<b>Pulse Width Modulation:</b> Output 0.
PWM1	5	O	<b>Pulse Width Modulation:</b> Output 1.
EW	6	I	<b>Enable Watchdog Timer:</b> Enable for T3 watchdog timer and disable power-down mode.
P0.0–P0.7	57–50	I/O	<b>Port 0:</b> Port 0 is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application it uses strong internal pull-ups when emitting 1s.
P1.0–P1.7	16–23	I/O	<b>Port 1:</b> 8-bit I/O port. Alternate functions include: (P1.0–P1.7): Quasi-bidirectional port pins.
	16–23	I/O	<b>CT0I–CT3I (P1.0–P1.3):</b> Capture timer input signals for timer T2.
	16–19	I/O	<b>T2 (P1.4):</b> T2 event input
	20	I	<b>RT2 (P1.5):</b> T2 timer reset signal. Rising edge triggered.
P2.0–P2.7	21	I	
P2.0–P2.7	39–46	I/O	<b>Port 2:</b> 8-bit quasi-bidirectional I/O port. Alternate function: High-order address byte for external memory (A08–A15).
P3.0–P3.7	24–31	I/O	<b>Port 3:</b> 8-bit quasi-bidirectional I/O port. Alternate functions include: <b>RxD (P3.0):</b> Serial input port. <b>TxD (P3.1):</b> Serial output port. <b>INT0 (P3.2):</b> External interrupt. <b>INT1 (P3.3):</b> External interrupt. <b>T0 (P3.4):</b> Timer 0 external input. <b>T1 (P3.5):</b> Timer 1 external input. <b>WR (P3.6):</b> External data memory write strobe. <b>RD (P3.7):</b> External data memory read strobe.
P4.0–P4.7	7–14	I/O	<b>Port 4:</b> 8-bit quasi-bidirectional I/O port. Alternate functions include: <b>CMSR0–CMSR5 (P4.0–P4.5):</b> Timer T2 compare and set/reset outputs on a match with timer T2.
	7–12	O	<b>CMT0, CMT1 (P4.6, P4.7):</b> Timer T2 compare and toggle outputs on a match with timer T2.
P5.0–P5.7	13, 14	O	
P5.0–P5.7	68–62, 1	I	<b>Port 5:</b> 8-bit input port. <b>ADC0–ADC7 (P5.0–P5.7):</b> Alternate function: Eight input channels to ADC.
RST	15	I/O	<b>Reset:</b> Input to reset the 87C552. It also provides a reset pulse as output when timer T3 overflows.
XTAL1	35	I	<b>Crystal Input 1:</b> Input to the inverting amplifier that forms the oscillator, and input to the internal clock generator. Receives the external clock signal when an external oscillator is used.
XTAL2	34	O	<b>Crystal Input 2:</b> Output of the inverting amplifier that forms the oscillator. Left open-circuit when an external clock is used.
V <sub>SS</sub>	36, 37	I	<b>Digital ground.</b>
PSEN	47	O	<b>Program Store Enable:</b> Active-low read strobe to external program memory.
ALE	48	O	<b>Address Latch Enable:</b> Latches the low byte of the address during accesses to external memory. It is activated every six oscillator periods. During an external data memory access, one ALE pulse is skipped. ALE can drive up to eight LS TTL inputs and handles CMOS inputs without an external pull-up.
E <sub>A</sub>	49	I	<b>External Access:</b> When E <sub>A</sub> is held at TTL level high, the CPU executes out of the internal program ROM provided the program counter is less than 8192. When E <sub>A</sub> is held at TTL low level, the CPU executes out of external program memory. E <sub>A</sub> is not allowed to float.
AV <sub>REF</sub> –	58	I	<b>Analog to Digital Conversion Reference Resistor:</b> Low-end.
AV <sub>REF</sub> +	59	I	<b>Analog to Digital Conversion Reference Resistor:</b> High-end.
AV <sub>SS</sub>	60	I	<b>Analog Ground</b>
AV <sub>DD</sub>	61	I	<b>Analog Power Supply</b>

## NOTE:

- To avoid "latch-up" effect at power-on, the voltage on any pin at any time must not be higher or lower than V<sub>DD</sub> + 0.5V or V<sub>SS</sub> – 0.5V, respectively.

Single-chip 8-bit microcontroller

80C562/83C562

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator, as shown in the logic symbol.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

RESET

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. To ensure a good power-on reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few milliseconds) plus two machine cycles. At power-on, the voltage on V<sub>DD</sub> and RST must come up at the same time for a proper start-up.

IDLE MODE

In the idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers

remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

POWER-DOWN MODE

In the power-down mode, the oscillator is stopped and the instruction to invoke power-down is the last instruction executed. Only the contents of the on-chip RAM are preserved. A hardware reset is the only way to terminate the power-down mode. The control bits for the reduced power modes are in the special function register PCON. Table 1 shows the state of the I/O ports during low current operating modes.

Table 1. External Pin Status During Idle and Power-Down Modes

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3	PORT 4	PWM0/PWM1
Idle	Internal	1	1	Data	Data	Data	Data	Data	High
Idle	External	1	1	Float	Data	Address	Data	Data	High
Power-down	Internal	0	0	Data	Data	Data	Data	Data	High
Power-down	External	0	0	Float	Data	Data	Data	Data	High

ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>

PARAMETER	RATING	UNIT
Voltage on any other pin to V <sub>SS</sub>	−0.5 to +6.5	V
Input, output DC current on any single I/O pin	5.0	mA
Power dissipation (based on package heat transfer limitations, not device power consumption)	1.0	W
Storage temperature range	−65 to +150	°C

NOTES:

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
- This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
- Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V<sub>SS</sub> unless otherwise noted.

## Single-chip 8-bit microcontroller

## 80C562/83C562

## DC ELECTRICAL CHARACTERISTICS

 $V_{SS}, AV_{SS} = 0V$ 

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			MIN	MAX	
$V_{DD}$	Supply voltage PCB8XC562 PCF8XC562 PCA8XC562		4.0 4.0 4.5	6.0 6.0 5.5	V
$I_{DD}$	Supply current operating: PCB8XC562 PCF8XC562 PCA8XC562	See notes 1 and 2 $f_{osc} = 16MHz$ $f_{osc} = 12MHz$ $f_{osc} = 12MHz$		45 34 30	mA
$I_{ID}$	Idle mode: PCB8XC562 PCF8XC562 PCA8XC562	See notes 1 and 3 $f_{osc} = 16MHz$ $f_{osc} = 12MHz$ $f_{osc} = 12MHz$		10 8 7	mA
$I_{PD}$	Power-down current:  PCB8XC562 PCF8XC562 PCA8XC562	See notes 1 and 4; $2V < V_{PD} < V_{DD} \text{ max}$		50 50 100	$\mu A$
<b>Inputs</b>					
$V_{IL}$	Input low voltage, except EA		-0.5	$0.2V_{DD}-0.1$	V
$V_{IL1}$	Input low voltage to EA		-0.5	$0.2V_{DD}-0.3$	V
$V_{IH}$	Input high voltage, except XTAL1, RST		$0.2V_{DD}+0.9$	$V_{DD}+0.5$	V
$V_{IH1}$	Input high voltage, XTAL1, RST		$0.7V_{DD}$	$V_{DD}+0.5$	V
$I_{IL}$	Logical 0 input current, ports 1, 2, 3, 4	$V_{IN} = 0.45V$		-50	$\mu A$
$I_{TL}$	Logical 1-to-0 transition current, ports 1, 2, 3, 4	See note 5		-650	$\mu A$
$\pm I_{IL1}$	Input leakage current, port 0, EA, STADC, EW	$0.45V < V_I < V_{DD}$		10	$\mu A$
<b>Outputs</b>					
$V_{OL}$	Output low voltage, ports 1, 2, 3, 4	$I_{OL} = 1.6mA^6$		0.45	V
$V_{OL1}$	Output low voltage, port 0, ALE, PSEN, PWM0, PWM1	$I_{OL} = 3.2mA^6$		0.45	V
$V_{OH}$	Output high voltage, ports 1, 2, 3, 4	$V_{DD} + 5V \pm 10\%$ $-I_{OH} = 60\mu A$ $-I_{OH} = 25\mu A$ $-I_{OH} = 10\mu A$	2.4 $0.75V_{DD}$ $0.9V_{DD}$	V V V	
$V_{OH1}$	Output high voltage (port 0 in external bus mode, ALE, PSEN, PWM0, PWM1) <sup>7</sup>	$V_{DD} + 5V \pm 10\%$ $-I_{OH} = 400\mu A$ $-I_{OH} = 150\mu A$ $-I_{OH} = 40\mu A$	2.4 $0.75V_{DD}$ $0.9V_{DD}$	V V V	
$V_{OH2}$	Output high voltage (RST)	$-I_{OH} = 400\mu A$ $-I_{OH} = 120\mu A$	2.4 $0.8V_{DD}$	V V	
$R_{RST}$	Internal reset pull-down resistor		50	150	k $\Omega$
$C_{IO}$	Pin capacitance	Test freq = 1MHz, $T_{amb} = 25^\circ C$		10	pF

## Single-chip 8-bit microcontroller

## 80C562/83C562

## DC ELECTRICAL CHARACTERISTICS (Continued)

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			MIN	MAX	
Analog Inputs					
AV <sub>DD</sub>	Analog supply voltage: PCB8XC562 PCF8XC562 PCA8XC562	AV <sub>DD</sub> = V <sub>DD</sub> ±0.2V AV <sub>DD</sub> = V <sub>DD</sub> ±0.2V AV <sub>DD</sub> = V <sub>DD</sub> ±0.2V	4.0 4.0 4.5	6.0 6.0 5.5	V V V
AI <sub>DD</sub>	Analog supply current: operating:	Port 5 = 0 to AV <sub>DD</sub>		1.2	mA
AI <sub>ID</sub>	Idle mode: PCB8XC562 PCF8XC562 PCA8XC562			50 50 100	μA μA μA
AI <sub>PD</sub>	Power-down mode: PCB8XC562 PCF8XC562 PCA8XC562	2V < AV <sub>PD</sub> < AV <sub>DD</sub> max		50 50 100	μA μA μA
AV <sub>IN</sub>	Analog input voltage		AV <sub>SS</sub> -0.2	AV <sub>DD</sub> +0.2	V
AV <sub>REF</sub>	Reference voltage: AV <sub>REF-</sub> AV <sub>REF+</sub>		AV <sub>SS</sub> -0.2	AV <sub>DD</sub> +0.2	V V
R <sub>REF</sub>	Resistance between AV <sub>REF+</sub> and AV <sub>REF-</sub>		5	25	kΩ
C <sub>IA</sub>	Analog input capacitance			15	pF
t <sub>ADS</sub>	Sampling time			6t <sub>CY</sub>	μs
t <sub>ADC</sub>	Conversion time (including sampling time)			24t <sub>CY</sub>	μs
DL <sub>e</sub>	Differential non-linearity <sup>8, 9, 10</sup>			±1	LSB
IL <sub>e</sub>	Integral non-linearity <sup>8, 11</sup>			±1	LSB
OS <sub>e</sub>	Offset error <sup>8, 12</sup>			±1	LSB
G <sub>e</sub>	Gain error <sup>8, 13</sup>			0.4	%
M <sub>CTC</sub>	Channel to channel matching			±1	LSB
C <sub>t</sub>	Crosstalk between inputs of port 5 <sup>14</sup>	0–100kHz		-60	dB

## NOTES:

- See Figures 8 through 12 for I<sub>DD</sub> test conditions.
- The operating supply current is measured with all output pins disconnected; XTAL1 driven with t<sub>r</sub> = t<sub>f</sub> = 10ns; V<sub>IL</sub> = V<sub>SS</sub> + 0.5V; V<sub>IH</sub> = V<sub>DD</sub> - 0.5V; XTAL2 not connected; EA = RST = Port 0 = EW = V<sub>DD</sub>; STADC = V<sub>SS</sub>.
- The idle mode supply current is measured with all output pins disconnected; XTAL1 driven with t<sub>r</sub> = t<sub>f</sub> = 10ns; V<sub>IL</sub> = V<sub>SS</sub> + 0.5V; V<sub>IH</sub> = V<sub>DD</sub> - 0.5V; XTAL2 not connected; Port 0 = EW = V<sub>DD</sub>; EA = RST = STADC = V<sub>SS</sub>.
- The power-down current is measured with all output pins disconnected; XTAL2 not connected; Port 0 = EW = V<sub>DD</sub>; EA = RST = STADC = XTAL1 = V<sub>SS</sub>.
- Pins of ports 1, 2, 3, and 4 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when V<sub>IN</sub> is approximately 2V.
- Capacitive loading on ports 0 and 2 may cause spurious noise to be superimposed on the V<sub>OL</sub>s of ALE and ports 1 and 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100pF), the noise pulse on the ALE pin may exceed 0.8V. In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input. I<sub>OL</sub> can exceed these conditions provided that no single output sinks more than 5mA and no more than two outputs exceed the test conditions.
- Capacitive loading on ports 0 and 2 may cause the V<sub>OH</sub> on ALE and PSEN to momentarily fall below the 0.9V<sub>DD</sub> specification when the address bits are stabilizing.
- Conditions: AV<sub>REF-</sub> = 0V; AV<sub>DD</sub> = 5.0V; AV<sub>REF+</sub> = 5.12V. ADC is monotonic with no missing codes.
- The differential non-linearity (DL<sub>e</sub>) is the difference between the actual step width and the ideal step width. (See Figure 1.)
- The ADC is monotonic; there are no missing codes.
- The integral non-linearity (IL<sub>e</sub>) is the peak difference between the center of the steps of the actual and the ideal transfer curve after appropriate adjustment of gain and offset error. (See Figure 1.)
- The offset error (OS<sub>e</sub>) is the absolute difference between the straight line which fits the actual transfer curve (after removing gain error), and a straight line which fits the ideal transfer curve. (See Figure 1.)
- The gain error (G<sub>e</sub>) is the relative difference in percent between the straight line fitting the actual transfer curve (after removing offset error), and the straight line which fits the ideal transfer curve. Gain error is constant at every point on the transfer curve. (See Figure 1.)
- This should be considered when both analog and digital signals are simultaneously input to port 5.

## Single-chip 8-bit microcontroller

## 80C562/83C562

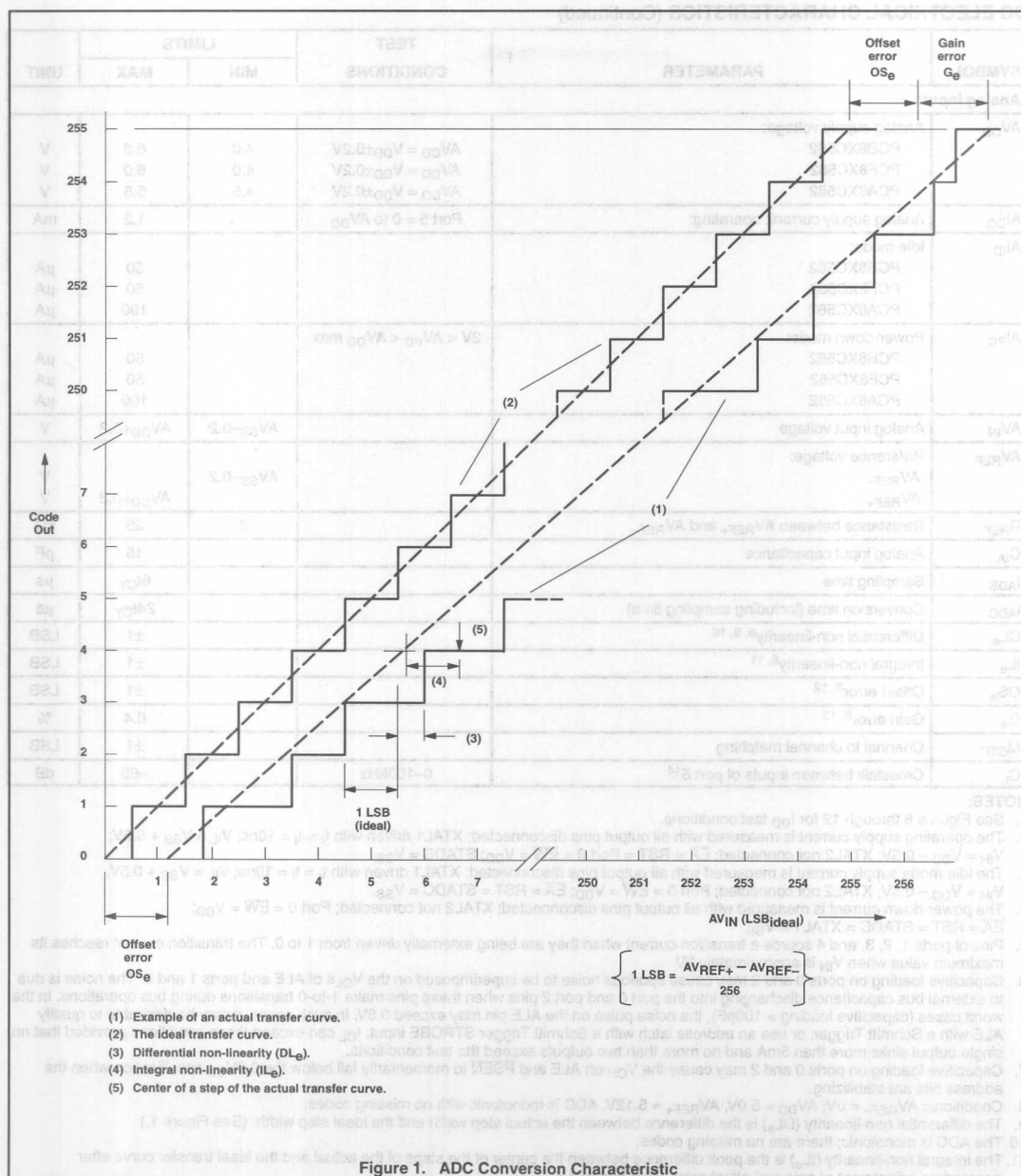


Figure 1. ADC Conversion Characteristic



## Single-chip 8-bit microcontroller

80C562/83C562

AC ELECTRICAL CHARACTERISTICS<sup>1, 2</sup>

SYMBOL	FIGURE	PARAMETER	12MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
$1/t_{CLCL}$	2	Oscillator frequency			1.2	16	MHz
$t_{LHLL}$	2	ALE pulse width	127		$2t_{CLCL}-40$		ns
$t_{AVLL}$	2	Address valid to ALE low	28		$t_{CLCL}-55$		ns
$t_{LLAX}$	2	Address hold after ALE low	48		$t_{CLCL}-35$		ns
$t_{LLIV}$	2	ALE low to valid instruction in		234	$4t_{CLCL}-100$		ns
$t_{LLPL}$	2	ALE low to PSEN low	43		$t_{CLCL}-40$		ns
$t_{PLPH}$	2	PSEN pulse width	205		$3t_{CLCL}-45$		ns
$t_{PLIV}$	2	PSEN low to valid instruction in		145		$3t_{CLCL}-105$	ns
$t_{PXIX}$	2	Input instruction hold after PSEN	0		0		ns
$t_{PXIZ}$	2	Input instruction float after PSEN		59		$t_{CLCL}-25$	ns
$t_{AVIV}$	2	Address to valid instruction in		312		$5t_{CLCL}-105$	ns
$t_{PLAZ}$	2	PSEN low to address float		10		10	ns
<b>Data Memory</b>							
$t_{AVLL}$	3, 4	Address valid to ALE low	43		$t_{CLCL}-35$		ns
$t_{RLRH}$	3	$\overline{RD}$ pulse width	400		$6t_{CLCL}-100$		ns
$t_{WLWH}$	4	$\overline{WR}$ pulse width	400		$6t_{CLCL}-100$		ns
$t_{RLDV}$	3	$\overline{RD}$ low to valid data in		252		$5t_{CLCL}-165$	ns
$t_{RHDX}$	3	Data hold after $\overline{RD}$	0		0		ns
$t_{RHDX}$	3	Data float after $\overline{RD}$		97		$2t_{CLCL}-70$	ns
$t_{LLDV}$	3	ALE low to valid data in		517		$8t_{CLCL}-150$	ns
$t_{AVDV}$	3	Address to valid data in		585		$9t_{CLCL}-165$	ns
$t_{LLWL}$	3, 4	ALE low to $\overline{RD}$ or $\overline{WR}$ low	200	300	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
$t_{AVWL}$	3, 4	Address valid to $\overline{WR}$ low or $\overline{RD}$ low	203		$4t_{CLCL}-130$		ns
$t_{QVWX}$	4	Data valid to $\overline{WR}$ transition	23		$t_{CLCL}-60$		ns
$t_{DW}$	4	Data before $\overline{WR}$	433		$7t_{CLCL}-150$		ns
$t_{WHQX}$	4	Data hold after $\overline{WR}$	33		$t_{CLCL}-50$		ns
$t_{RLAZ}$	3	$\overline{RD}$ low to address float		0		0	ns
$t_{WHLH}$	3, 4	$\overline{RD}$ or $\overline{WR}$ high to ALE high	43	123	$t_{CLCL}-40$	$t_{CLCL}+40$	ns
<b>External Clock</b>							
$t_{CHCX}$	5	High time <sup>3</sup>	20		20		ns
$t_{CLCX}$	5	Low time <sup>3</sup>	20		20		ns
$t_{CLCH}$	5	Rise time <sup>3</sup>		20		20	ns
$t_{CHCL}$	5	Fall time <sup>3</sup>		20		20	ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- These values are characterized but not 100% production tested.

Single-chip 8-bit microcontroller

80C562/83C562

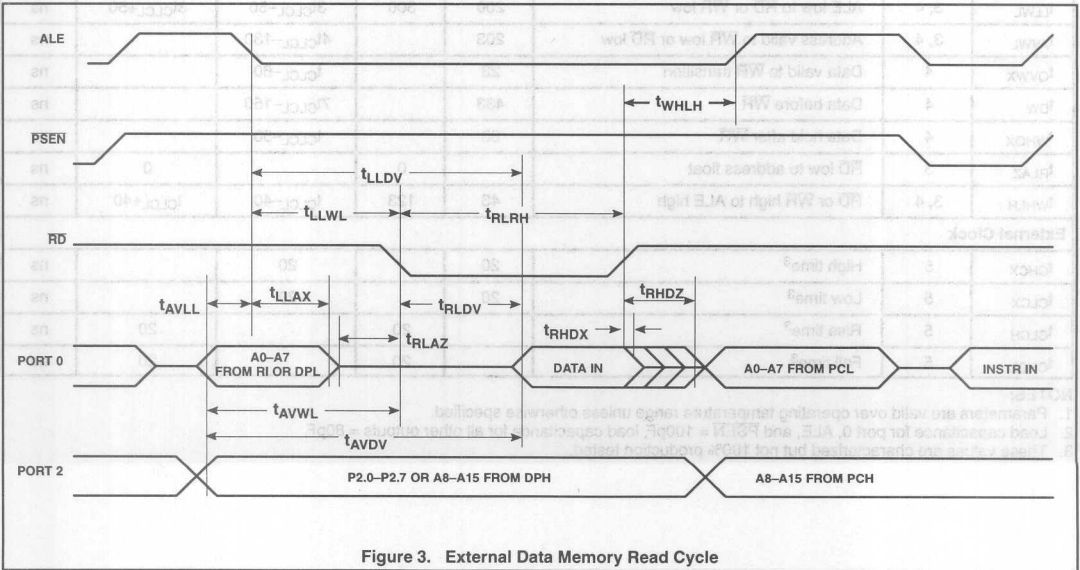
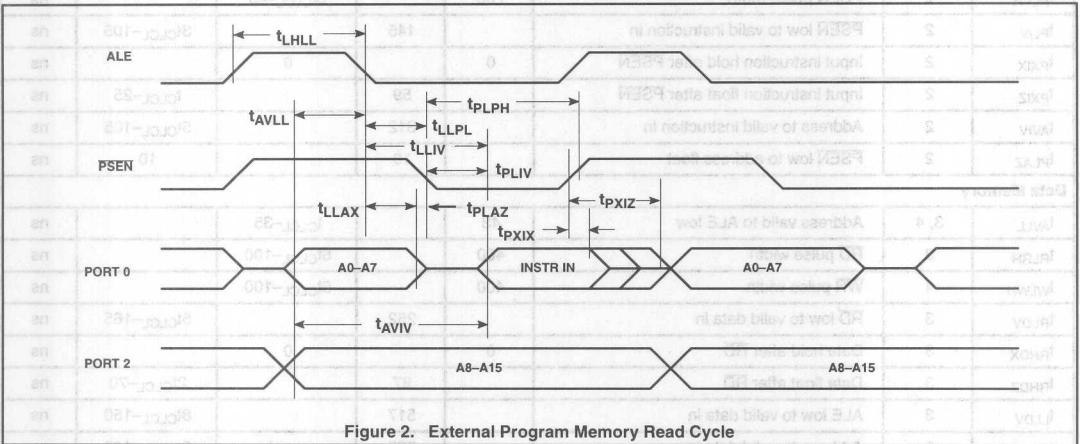
EXPLANATION OF THE AC SYMBOLS

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

- A – Address
- C – Clock
- D – Input data
- H – Logic level high
- I – Instruction (program memory contents)
- L – Logic level low, or ALE
- P – PSEN

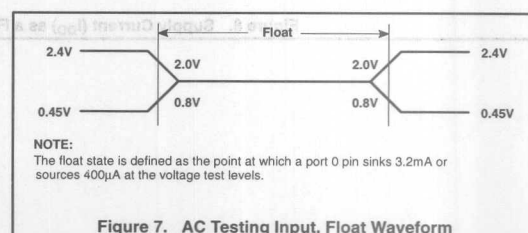
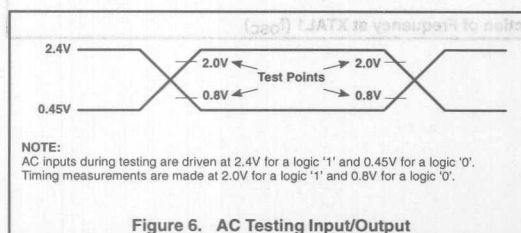
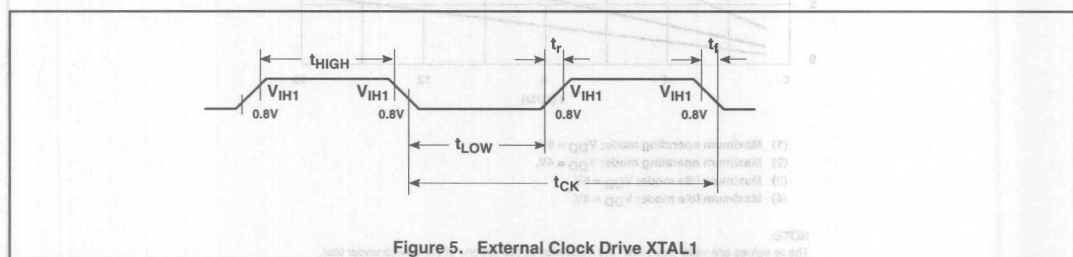
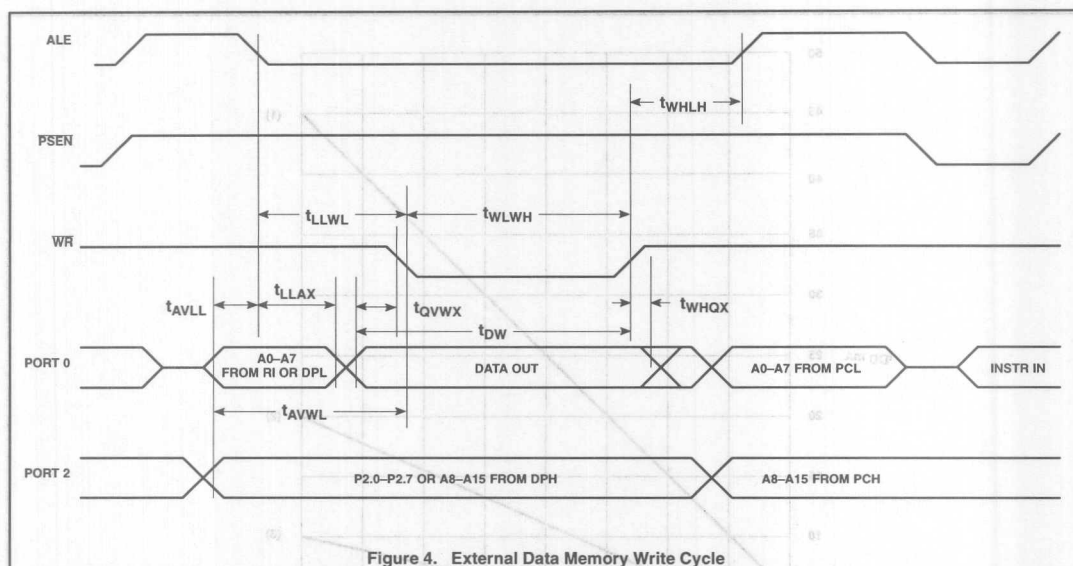
- Q – Output data
- R – RD signal
- t – Time
- V – Valid
- W – WR signal
- X – No longer a valid logic level
- Z – Float

Examples:  $t_{AVLL}$  = Time for address valid to ALE low.  
 $t_{LLPL}$  = Time for ALE low to PSEN low.



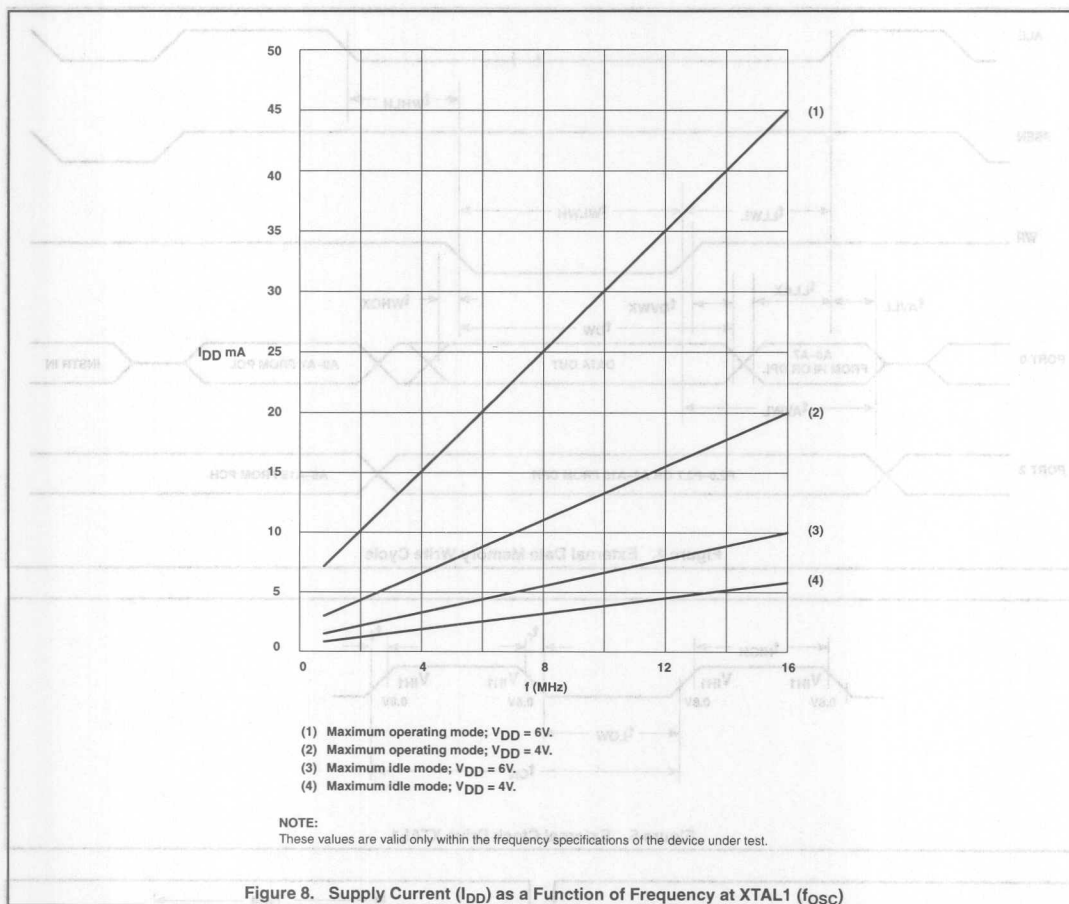
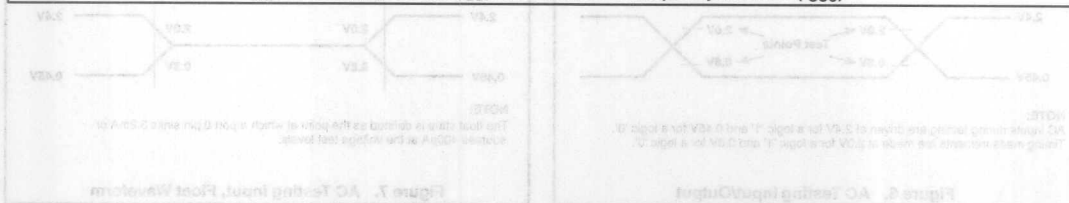
## Single-chip 8-bit microcontroller

## 80C562/83C562



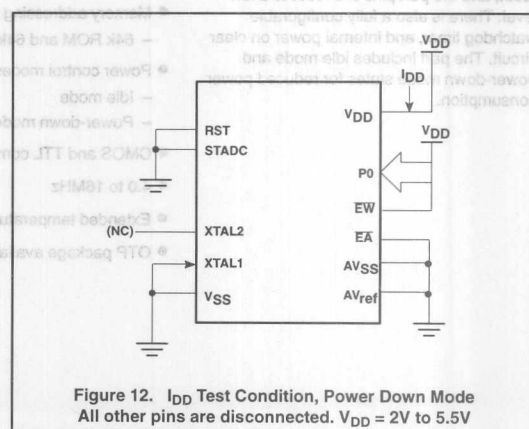
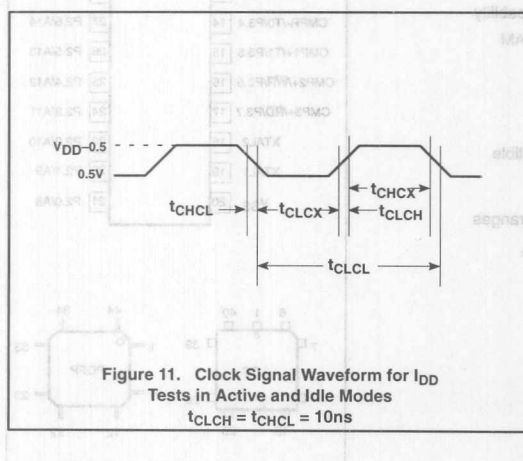
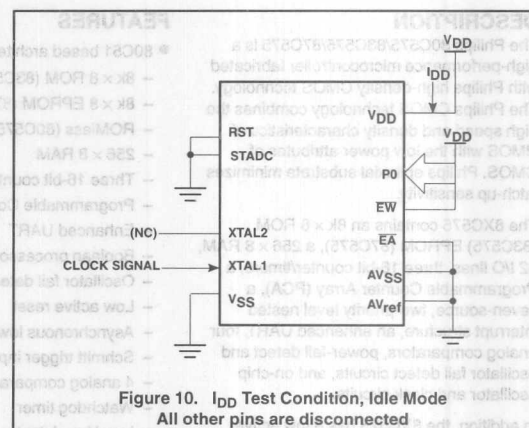
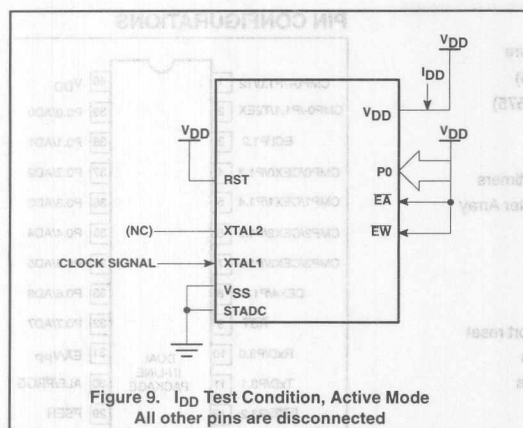
## Single-chip 8-bit microcontroller

## 80C562/83C562

Figure 8. Supply Current ( $I_{DD}$ ) as a Function of Frequency at XTAL1 ( $f_{osc}$ )

## Single-chip 8-bit microcontroller

## 80C562/83C562





## CMOS single-chip 8-bit microcontroller

80C575/83C575/87C575

## DESCRIPTION

The Philips 80C575/83C575/87C575 is a high-performance microcontroller fabricated with Philips high-density CMOS technology. The Philips CMOS technology combines the high speed and density characteristics of HMOS with the low power attributes of CMOS. Philips epitaxial substrate minimizes latch-up sensitivity.

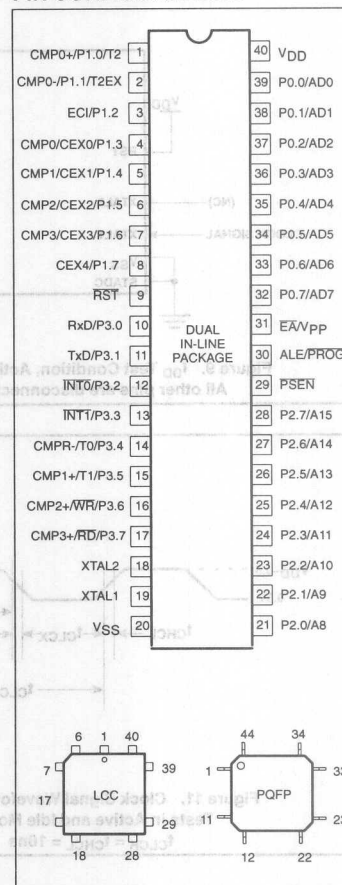
The 8XC575 contains an 8k × 8 ROM (83C575) EPROM (87C575), a 256 × 8 RAM, 32 I/O lines, three 16-bit counter/timers, a Programmable Counter Array (PCA), a seven-source, two-priority level nested interrupt structure, an enhanced UART, four analog comparators, power-fail detect and oscillator fail detect circuits, and on-chip oscillator and clock circuits.

In addition, the 8XC575 has a low active reset, and the port pins are reset to a low level. There is also a fully configurable watchdog timer, and internal power on clear circuit. The part includes idle mode and power-down mode states for reduced power consumption.

## FEATURES

- 80C51 based architecture
  - 8k × 8 ROM (83C575)
  - 8k × 8 EPROM (87C575)
  - ROMless (80C575)
  - 256 × 8 RAM
  - Three 16-bit counter/timers
  - Programmable Counter Array
  - Enhanced UART
  - Boolean processor
  - Oscillator fail detect
  - Low active reset
  - Asynchronous low port reset
  - Schmitt trigger inputs
  - 4 analog comparators
  - Watchdog timer
  - Low V<sub>CC</sub> detect
- Memory addressing capability
  - 64k ROM and 64k RAM
- Power control modes:
  - Idle mode
  - Power-down mode
- CMOS and TTL compatible
- 4.0 to 16MHz
- Extended temperature ranges
- OTP package available

## PIN CONFIGURATIONS



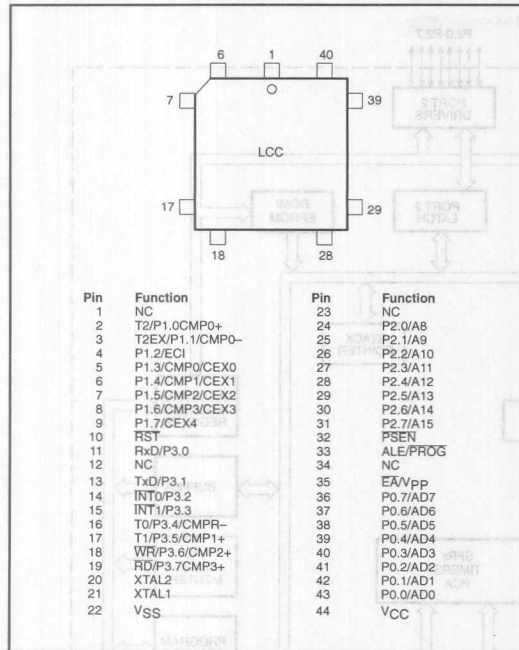
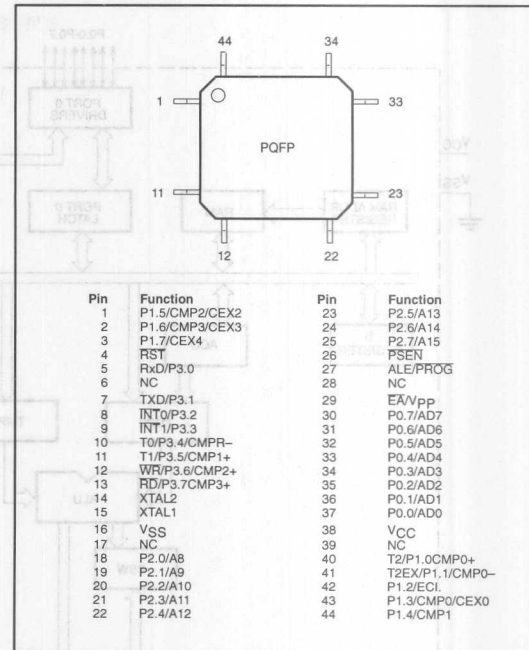
## ORDERING INFORMATION

ROMless	ROM	EPROM		TEMPERATURE RANGE °C AND PACKAGE <sup>1</sup>	FREQ (MHz)	DRAWING NUMBER
P80C575EBPN	P83C575EBPN	P87C575EBPN	OTP	0 to +70, 40-Pin Plastic Dual In-line Package	16	0415C
P80C575EBAA	P83C575EBAA	P87C575EBAA	OTP	0 to +70, 44-Pin Plastic Leaded Chip Carrier	16	0403G
		P87C575EBFFA	UV	0 to +70, 40-Pin Ceramic Dual In-line Package	16	0590B
		P87C575EBLKA	UV	0 to +70, 44-Pin Ceramic Leaded Chip Carrier	16	1472A
P80C575EHPN	P83C575EHPN	P87C575EHPN	OTP	-40 to +125, 40-Pin Plastic Dual In-line Package	16	0415C
P80C575EHAA	P83C575EHAA	P87C575EHAA	OTP	-40 to +125, 44-Pin Plastic Leaded Chip Carrier	16	0403G
		P87C575EHFFA	UV	-40 to +125, 40-Pin Ceramic Dual In-line Package	16	0590B
		P87C575EHLKA	UV	-40 to +125, 44-Pin Ceramic Leaded Chip Carrier	16	1472A
P80C575EBBB	P83C575EBBB	P87C575EBBB	OTP	0 to +70, 44-Pin Plastic Quad Flat Pack	16	1118D

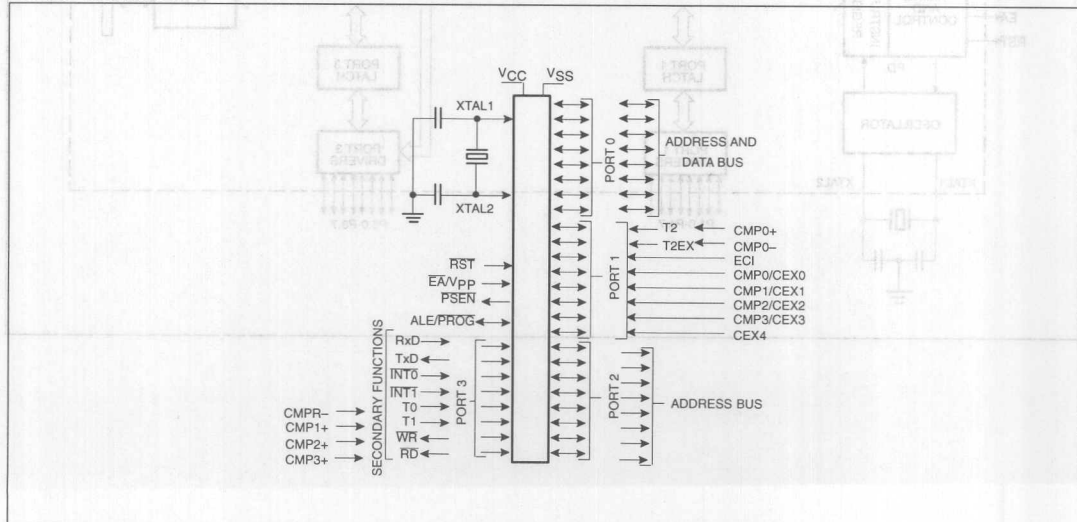
## NOTE:

1. OTP - One Time Programmable EPROM. UV - Erasable EPROM

## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575

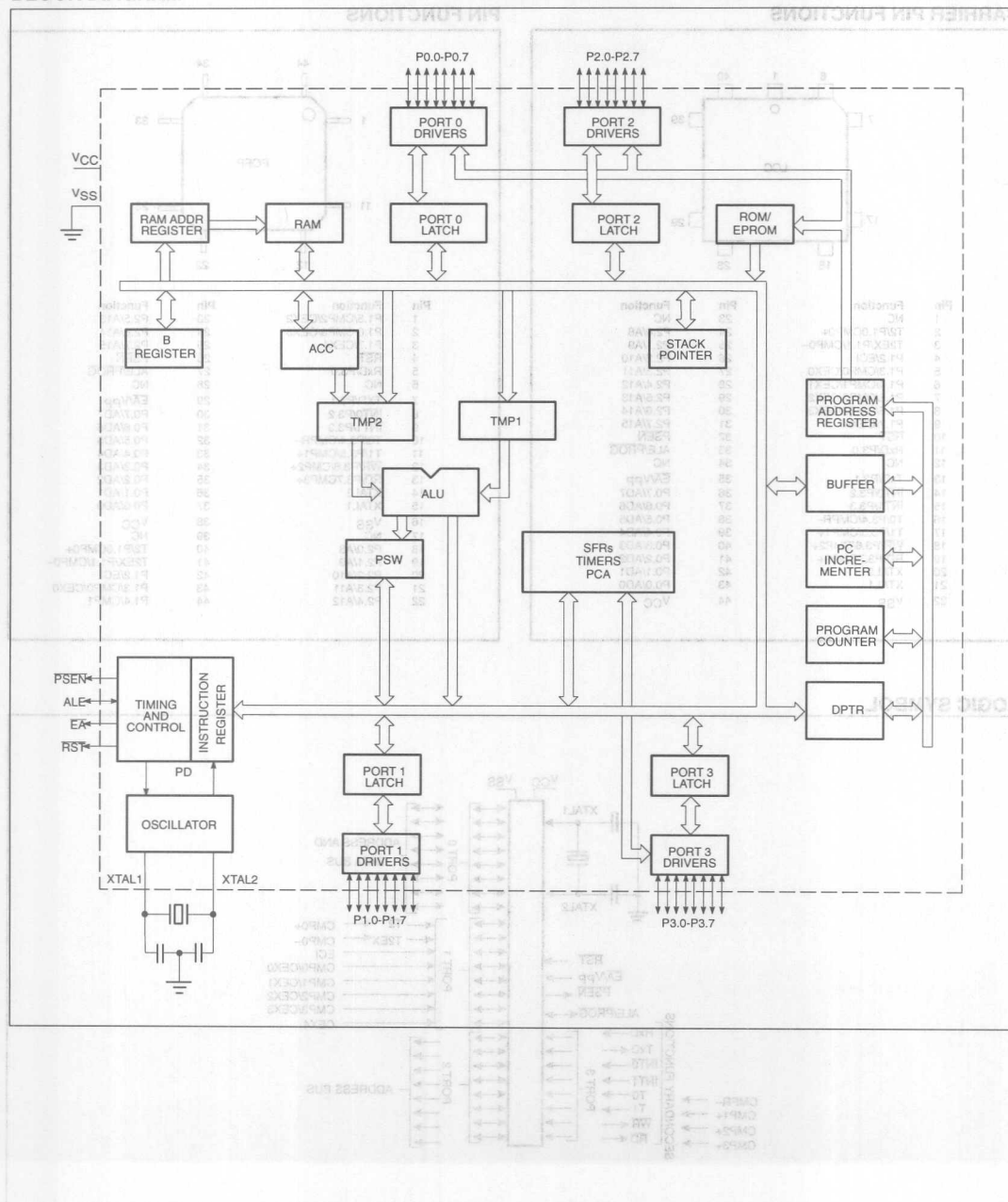
CERAMIC AND PLASTIC LEADED CHIP  
CARRIER PIN FUNCTIONSPLASTIC QUAD FLAT PACK  
PIN FUNCTIONS

## LOGIC SYMBOL



## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575

## BLOCK DIAGRAM



## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575

## PIN DESCRIPTIONS

MNEMONIC	PIN NUMBER			TYPE	NAME AND FUNCTION
	DIP	LCC	QFP		
V <sub>SS</sub>	20	22	16	I	<b>Ground:</b> 0V reference.
V <sub>CC</sub>	40	44	38	I	<b>Power Supply:</b> This is the power supply voltage for normal, idle, and power-down operation.
P0.0-0.7	39-32	43-36	37-30	I/O	<b>Port 0:</b> Port 0 is an open-drain bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s. Port 0 also receives code bytes during EPROM programming and outputs code bytes during program verification. External pull-ups are required during program verification. During reset, port 0 will be asynchronously driven low and will remain low until written to by software. All port 0 pins have Schmitt trigger inputs with 200mV hysteresis. A weak pulldown on port 0 guarantees positive leakage current (see DC Electrical Characteristics: I <sub>L1</sub> ).
P1.0-P1.7	1-8	2-9	40-44 1-3	I/O	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port. Port 1 pins have internal pull-ups such that pins that have 1s written to them can be used as inputs but will source current when externally pulled low (see DC Electrical Characteristics: I <sub>L1</sub> ). Port 1 receives the low-order address byte during program memory verification and EPROM programming. During reset, port 1 will be asynchronously driven low and will remain low until written to by software. All port 1 pins have Schmitt trigger inputs with 50mV hysteresis. Port 1 pins also serve alternate functions as follows:
	1	2	40	I/O	<b>P1.0 T2</b> Timer 2 external I/O
					<b>CMP0+</b> Comparator 0 positive input
	2	3	41	I	<b>P1.1 T2EX</b> Timer 2 capture input
					<b>CMP0-</b> Comparator 0 negative input
	3	4	42	I	<b>P1.2 ECI</b> PCA count input
	4	5	43	I/O	<b>P1.3 CEX0</b> PCA module 0 external I/O
					<b>CMP0</b> Comparator 0 output
	5	6	44	I/O	<b>P1.4 CEX1</b> PCA module 1 external I/O
					<b>CMP1</b> Comparator 1 output
	6	7	1	I/O	<b>P1.5 CEX2</b> PCA module 2 external I/O
					<b>CMP2</b> Comparator 2 output
	7	8	2	I/O	<b>P1.6 CEX3</b> PCA module 3 external I/O
					<b>CMP3</b> Comparator 3 output
	8	9	3	I/O	<b>P1.7 CEX4</b> PCA module 4 external I/O
P2.0-P2.7	21-28	24-31	18-25	I/O	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them can be used as inputs, but will source current when externally pulled low (see DC Electrical Characteristics: I <sub>L1</sub> ). Port 2 emits the high-order address byte during accesses to external program and data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. Port 2 receives the high-order address byte during program verification and EPROM programming. During reset, port 2 will be asynchronously driven low and will remain low until written to by software. Port 2 can be made open drain by writing to the P2OD register (AIH). In open drain mode, weak pulldowns on port 2 guarantee positive leakage current (see DC Electrical Characteristics I <sub>L1</sub> ).
P3.0-P3.7	10-17	11, 13-19	5, 7-13	I/O	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins except P3.1 that have 1s written to them can be used as inputs but will source current when externally pulled low (see DC Electrical Characteristics: I <sub>L1</sub> ). P3.1 will be a high impedance pin except while transmitting serial data, in which case the strong pull-up will remain on continuously when outputting a 1 level. The P3.1 output drive level when transmitting can be set to one of two levels by the writing to the P3.1 register bit. During reset all pins (except P3.1) will be asynchronously driven low and will remain low until written to by software. All port 3 pins have Schmitt trigger inputs with 200mV hysteresis, except P3.2 and P3.3, which have 50mV hysteresis. Port 3 pins serve alternate functions as follows:

## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575

## PIN DESCRIPTIONS (Continued)

MNEMONIC	PIN NUMBER			TYPE	NAME AND FUNCTION
	DIP	LCC	QFP		
	10	11	5	I	<b>Port 3: (continued)</b>
	11	13	7	O	<b>P3.0 RxD</b> Serial receive port
	12	14	8	I	<b>P3.1 TxD</b> Serial transmit port enabled only when transmitting serial data
	13	15	9	I	<b>P3.2 INT0</b> External interrupt 0
	14	16	10	I	<b>P3.3 INT1</b> External interrupt 1
	15	17	11	I	<b>P3.4 T0</b> Timer/counter 0 input
	16	18	12	O	<b>CMPR-</b> Common - reference to comparators 1, 2, 3
	17	19	13	O	<b>P3.5 T1</b> Timer/counter 1 input
					<b>CMP1+</b> Comparator 1 positive input
					<b>P3.6 WR</b> External data memory write strobe
					<b>CMP2+</b> Comparator 2 positive input
					<b>P3.7 RD</b> External data memory read strobe
					<b>CMP3+</b> Comparator 3 positive input
RST	9	10	4	I	<b>Reset:</b> A low on this pin asynchronously resets all port pins to a low state except P3.1. The pin must be held low with the oscillator running for 24 oscillator cycles to initialize the internal registers. An internal diffused resistor to V <sub>CC</sub> permits a power on reset using only an external capacitor to V <sub>SS</sub> . RST has a Schmitt trigger input stage to provide additional noise immunity with a slow rising input voltage.
ALE/PROG	30	33	27	I/O	<b>Address Latch Enable/Program Pulse:</b> Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory. ALE is switched off if the bit 0 in the AUXR register (8EH) is set. This pin is also the program pulse input (PROG) during EPROM programming.
PSEN	29	32	26	O	<b>Program Store Enable:</b> The read strobe to external program memory. When the device is executing code from the external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory. PSEN is not activated during fetches from internal program memory.
EA/V <sub>PP</sub>	31	35	29	I	<b>External Access Enable/Programming Supply Voltage:</b> EA must be externally held low to enable the device to fetch code from external program memory locations 0000H to 1FFFFH. If EA is held high, the device executes from internal program memory unless the program counter contains an address greater than 1FFFFH. This pin also receives the 12.75V programming supply voltage (V <sub>PP</sub> ) during EPROM programming.
XTAL1	19	21	15	I	<b>Crystal 1:</b> Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	20	14	O	<b>Crystal 2:</b> Output from the inverting oscillator amplifier.



## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575

Table 1. 87C575 Special Function Registers (Continued)

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB				LSB				
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
AUXR#	Auxiliary	8EH	—	—	—	—	—	—	LO	AO	xxxxxx00B
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
CCAP0H#	Module 0 Capture High	FAH									xxxxxxx0B
CCAP1H#	Module 1 Capture High	FBH									xxxxxxx0B
CCAP2H#	Module 2 Capture High	FCH									xxxxxxx0B
CCAP3H#	Module 3 Capture High	FDH									xxxxxxx0B
CCAP4H#	Module 4 Capture High	FEH									xxxxxxx0B
CCAP0L#	Module 0 Capture Low	EAH									xxxxxxx0B
CCAP1L#	Module 1 Capture Low	EBH									xxxxxxx0B
CCAP2L#	Module 2 Capture Low	ECH									xxxxxxx0B
CCAP3L#	Module 3 Capture Low	EDH									xxxxxxx0B
CCAP4L#	Module 4 Capture Low	EEH									xxxxxxx0B
CCAPM0#	Module 0 Mode	DAH	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM1#	Module 1 Mode	DBH	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM2#	Module 2 Mode	DCH	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM3#	Module 3 Mode	DDH	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM4#	Module 4 Mode	DEH	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCON*# CH# CL#	PCA Counter Control	D8H	DF	DE	DD	DC	DB	DA	D9	D8	00x00000B 00H 00H
	PCA Counter High	F9H	CF	CR	—	CCF4	CCF3	CCF2	CCF1	CCF0	
	PCA Counter Low	E9H	—	—	—	—	—	—	—	—	
CMOD#	PCA Counter Mode	D9H	CIDL	WDTE	—	—	—	CPS1	CPS0	ECF	00xxx000B
CMP*# CMPE#	Comparator	E8H	EF	EE	ED	EC	EB	EA	E9	E8	00H 00H
	Comparator Enable	91H	EC3DP	EC2DP	EC1DP	EC0DP	C3RO	C2RO	C1RO	C0RO	
				EC3TDC	EC2TDC	EC1TDC	EC0TDC	EC3OD	EC2OD	EC1OD	
DPTR: DPH DPL	Data Pointer (2 bytes)										00H 00H
	Data Pointer High	83H									
	Data Pointer Low	82H									
IE*	Interrupt Enable	A8H	AF	AE	AD	AC	AB	AA	A9	A8	00H
			EA	EC	ET2	ES	ET1	EX1	ET0	EX0	
			BF	BE	BD	BC	BB	BA	B9	B8	
IP*	Interrupt Priority	B8H	—	PPC	PT2	PS	PT1	PX1	PT0	PX0	x0000000B
			87	86	85	84	83	82	81	80	
			AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	
P0*	Port 0	80H	97	96	95	94	93	92	91	90	00H
			CEX4	CEX3	CEX2	CEX1	CEX0	EXI	T2EX	T2	
			A7	A6	A5	A4	A3	A2	A1	A0	
P1*	Port 1	90H	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8	00H
			B7	B6	B5	B4	B3	B2	B1	B0	
			RD	WR	T1	T0	INT1	INT0	TxD	RxD	
P2*	Port 2	A0H									
P3*	Port 3	B0H									

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.

## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575

Table 1. 87C575 Special Function Registers (Continued)

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION																RESET VALUE
			MSB								LSB								
P2OD#	Port 2 Pullup Disable	A1H																	00H
PCON	Power Control	87H	SMOD1	SMOD0	OSF1	POF1	LVF1	GF0	PD	IDL								00xxx000B	
PSW*	Program Status Word	D0H	D7	D6	D5	D4	D3	D2	D1	D0								00H	
RACAP2H#	Timer 2 Capture High	CBH																	00H
RACAP2L#	Timer 2 Capture Low	CAH																	00H
SADDR#	Slave Address	A9H																	00H
SADEN#	Slave Address Mask	B9H																	00H
SBUF	Serial Data Buffer	99H																	xxxxxxx0B
SCON*	Serial Control	98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI							00H		
SP	Stack Pointer	81H																	07H
TCON*	Timer Control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IE0							00H		
T2CON*	Timer 2 Control	C8H	CF	CE	CD	CC	CB	CA	C9	C8							00H		
T2MOD#	Timer 2 Mode Control	C9H	-	-	-	-	-	-	-	DCEN							xxxxxxx0B		
TH0	Timer High 0	8CH																	00H
TH1	Timer High 1	8DH																	00H
TH2#	Timer High 2	CDH																	00H
TL0	Timer Low 0	8AH																	00H
TL1	Timer Low 1	8BH																	00H
TL2#	Timer Low 2	CCH																	00H
TMOD	Timer Mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0							00H		
WDCON*#	Watchdog Timer Control	C0H	PRE2	PRE1	PRE0	LVRE	OFRE	WDRUN	WDT0F	WDMOD							11111101B		
WDL#	Watchdog Timer Reload	C1H																	00H
WFEED1#	Watchdog Feed 1	C2H																	xxH
WFEED2#	Watchdog Feed 2	C3H																	xxH

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.

1. Reset value depends on reset source.

## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575

**POWER ON CLEAR/  
POWER ON FLAG**

An on-chip Power On Detect Circuit resets the 8XC575 and sets the Power Off Flag (PCON.4) on power up or if  $V_{CC}$  drops to zero momentarily. The POF can only be cleared by software. The RST pin is not driven by the power on detect circuit. The POF can be read by software to determine that a power failure has occurred and can also be set by software.

**LOW VOLTAGE DETECT**

An on-chip Low Voltage Detect circuit sets the Low Voltage Flag (PCON.3) if  $V_{CC}$  drops below  $V_{LOW}$  (see DC Electrical Characteristics) and resets the 8XC575 if the Low Voltage Reset Enable bit (WDCON.4) is set. If the LVRE is cleared, the reset is disabled but LVF will still be set if  $V_{CC}$  is low. The RST pin is not driven by the low voltage detect circuit. The LVF can be read by software to determine that  $V_{CC}$  was low. The LVF can be set or cleared by software.

**OSCILLATOR FAIL DETECT**

An on-chip Oscillator Fail Detect circuit sets the Oscillator Fail Flag (PCON.5) if the oscillator frequency drops below OSCF for one or more cycles (see AC Electrical Characteristics: OSCF) and resets the 8XC575 if the Oscillator Fail Reset Enable bit (WDCON.3) is set. If OFRE is cleared, the reset is disabled but OSF will still be set if the oscillator fails. The RST pin is not driven by the oscillator fail detect circuit. The OSF can be read by software to determine that an oscillator failure has occurred. The OSF can be set or cleared by software.

**LOW ACTIVE RESET**

One of the most notable features on this part is the low active reset. At this time this is the only 80C51 derivative available that has low active reset. This feature makes it easier to interface the 8XC575 into an application to accommodate the power-on and low voltage conditions that can occur. The low active reset operates exactly the same as high active reset with the exception that the part is put into the reset mode by applying a low level to the reset pin. For power-on reset it is also necessary to invert the power-on reset circuit; connecting the 8.2K resistor from the reset pin to  $V_{CC}$  and the 10 $\mu$ F capacitor from the reset pin to ground. Figure 1 shows all of the reset related circuitry.

When reset the port pins on the 87C575 are driven low asynchronously. This is different from all other 80C51 derivatives.

The 8XC575 also has Low voltage detection circuitry that will, if enabled, force the part to reset when  $V_{CC}$  (on the part) fails below a set level. Low Voltage Reset is enabled by a normal reset. Low Voltage Reset can be disabled by clearing LVRE (bit 4 in the WDCON SFR) then executing a watchdog feed sequence (ASH to WFEED1 followed immediately by 5A to WFEED2). In addition there is a flag (LVF) that is set if a low voltage condition is detected. The LVF flag is set even if the Low Voltage detection circuitry is disabled. Notice that the Low voltage detection circuitry does not drive the RST# pin so the LVF flag is the only way that the microcontroller can determine if it has been reset due to a low voltage condition.

The 8XC575 has an on-chip power-on detection circuit that sets the POF (PCON.4) flag on power up or if the  $V_{CC}$  level momentarily drops to 0V. This flag can be used to determine if the part is being started from a power-on (cold start) or if a reset has occurred due to another condition (warm start).

**TIMERS**

The 87C575 has four on-chip timers.

Timers 0 and 1 are identical in every way to Timers 0 and 1 on the 80C51.

Timer 2 on the 8XC575 is identical to the 80C52 Timer 2 (described in detail in the 80C52 overview) with the exception that it is an up or down counter. To configure the Timer to count down the DCEN bit in the T2MOD special function register must be set and a low level must be present on the T2EX pin (P1.1).

The Watchdog timer operation and implementation is the same as that for the 8XC550 (described in the 8XC550 overview) with the exception that the reset values of the WDCON and WDL special function registers have been changed. The changes in these registers cause the watchdog timer to be enabled with a timeout of  $98304 \times T_{OSC}$  when the part is reset. The watchdog can be disabled by executing a valid feed sequence and then clearing WDRUN (bit 2 in the WDCON SFR).

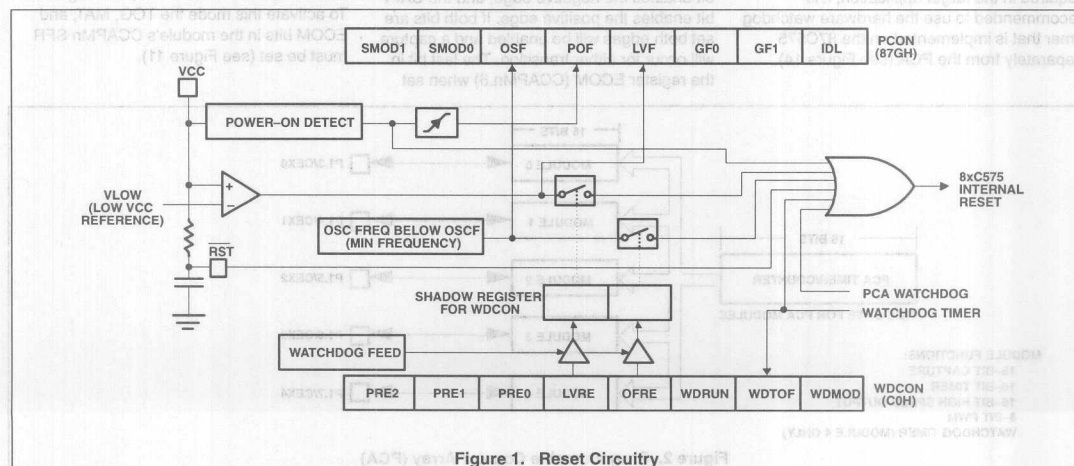


Figure 1. Reset Circuitry

## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575

**PROGRAMMABLE COUNTER ARRAY (PCA)**

The Programmable Counter Array is a special Timer that has five 16-bit capture/compare modules associated with it. Each of the modules can be programmed to operate in one of four modes: rising and/or falling edge capture, software timer, high-speed output, or pulse width modulator. Each module has a pin associated with it in port 1. Module 0 is connected to P1.3(CEX0), module 1 to P1.4(CEX1), etc.. The basic PCA configuration is shown in Figure 2.

The PCA timer is a common time base for all five modules and can be programmed to run at: 1/12 the oscillator frequency, 1/4 the oscillator frequency, the Timer 0 overflow, or the input on the ECI pin (P1.2). The timer count source is determined from the CPS1 and CPS0 bits in the CMOD SFR as follows (see Figure 3):

CPS1	CPS0	PCA Timer Count Source
0	0	1/12 oscillator frequency
0	1	1/4 oscillator frequency
1	0	Timer 0 overflow
1	1	External Input at ECI pin

In the CMOD SFR are three additional bits associated with the PCA. They are CIDL which allows the PCA to stop during idle mode, WDTE which enables or disables the watchdog function on module 4, and ECF which when set causes an interrupt and the PCA overflow flag CF (in the CCON SFR) to be set when the PCA timer overflows. These functions are shown in Figure 3.

The watchdog timer function is implemented in module 4 as implemented in other parts that have a PCA that are available on the market. However, if a watchdog timer is required in the target application, it is recommended to use the hardware watchdog timer that is implemented on the 87C575 separately from the PCA (see Figure 14).

The CCON SFR contains the run control bit for the PCA and the flags for the PCA timer (CF) and each module (refer to Figure 6). To run the PCA the CR bit (CCON.6) must be set by software. The PCA is shut off by clearing this bit. The CF bit (CCON.7) is set when the PCA counter overflows and an interrupt will be generated if the ECF bit in the CMOD register is set. The CF bit can only be cleared by software. Bits 0 through 4 of the CCON register are the flags for the modules (bit 0 for module 0, bit 1 for module 1, etc.) and are set by hardware when either a match or a capture occurs. These flags also can only be cleared by software. The PCA interrupt system shown in Figure 4.

Each module in the PCA has a special function register associated with it. These registers are: CCAPM0 for module 0, CCAPM1 for module 1, etc. (see Figure 7). The registers contain the bits that control the mode that each module will operate in. The ECCF bit (CCAPMn.0 where n=0, 1, 2, 3, or 4 depending on the module) enables the CCF flag in the CCON SFR to generate an interrupt when a match or compare occurs in the associated module. PWM (CCAPMn.1) enables the pulse width modulation mode. The TOG bit (CCAPMn.2) when set causes the CEX output associated with the module to toggle when there is a match between the PCA counter and the module's capture/compare register. The match bit MAT (CCAPMn.3) when set will cause the CCFn bit in the CCON register to be set when there is a match between the PCA counter and the module's capture/compare register.

The next two bits CAPN (CCAPMn.4) and CAPP (CCAPMn.5) determine the edge that a capture input will be active on. The CAPN bit enables the negative edge, and the CAPP bit enables the positive edge. If both bits are set both edges will be enabled and a capture will occur for either transition. The last bit in the register ECOM (CCAPMn.6) when set

enables the comparator function. Figure 8 shows the CCAPMn settings for the various PCA functions.

There are two additional registers associated with each of the PCA modules. They are CCAPnH and CCAPnL and these are the registers that store the 16-bit count when a capture occurs or a compare should occur. When a module is used in the PWM mode these registers are used to control the duty cycle of the output.

**PCA Capture Mode**

To use one of the PCA modules in the capture mode either one or both of the CCAPM bits CAPN and CAPP for that module must be set. The external CEX input for the module (on port 1) is sampled for a transition. When a valid transition occurs the PCA hardware loads the value of the PCA counter registers (CH and CL) into the module's capture registers (CCAPnL and CCAPnH). If the CCFn bit for the module in the CCON SFR and the ECCFn bit in the CCAPMn SFR are set then an interrupt will be generated. Refer to Figure 9.

**16-bit Software Timer Mode**

The PCA modules can be used as software timers by setting both the ECOM and MAT bits in the modules CCAPMn register. The PCA timer will be compared to the module's capture registers and when a match occurs an interrupt will occur if the CCFn (CCON SFR) and the ECCF (CCAPMn SFR) bits for the module are both set (see Figure 10).

**High Speed Output Mode**

In this mode the CEX output (on port 1) associated with the PCA module will toggle each time a match occurs between the PCA counter and the module's capture registers. To activate this mode the TOG, MAT, and ECOM bits in the module's CCAPMn SFR must be set (see Figure 11).

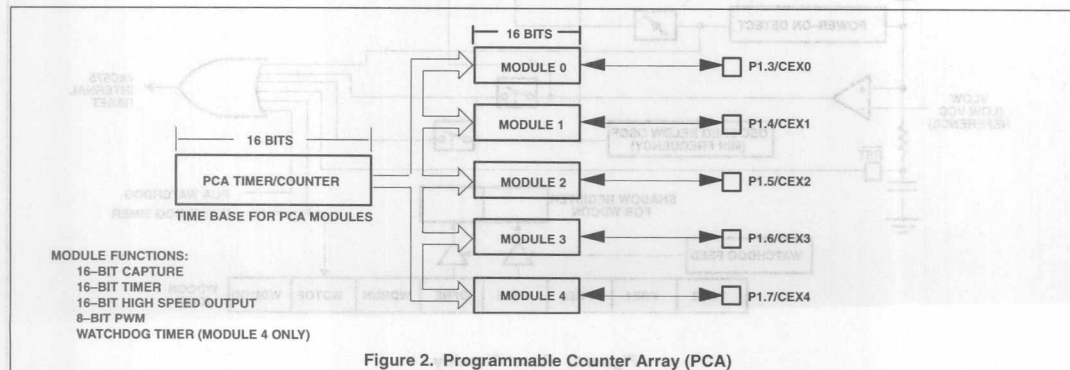


Figure 2. Programmable Counter Array (PCA)

## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575

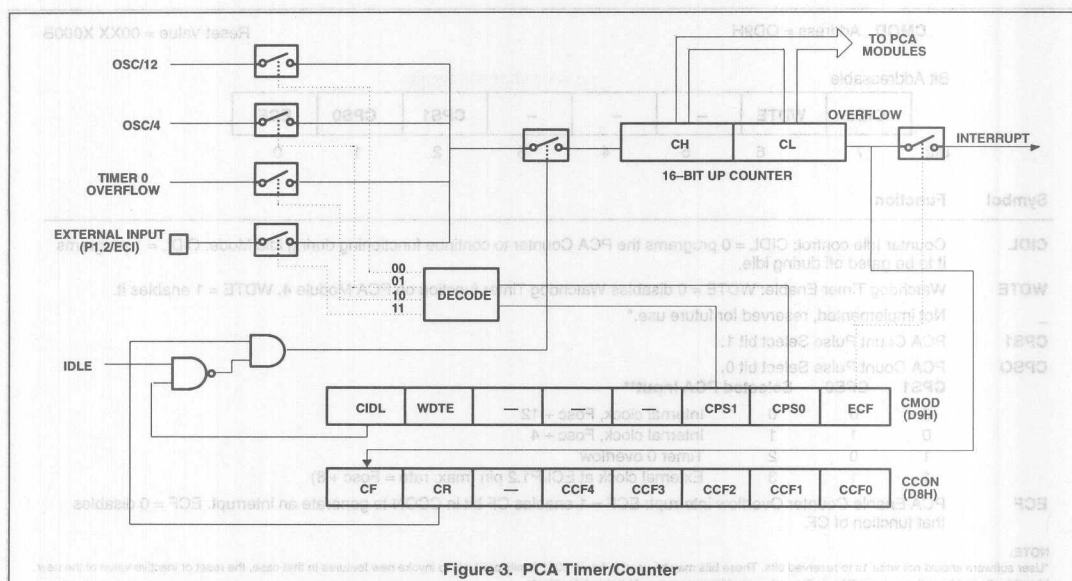
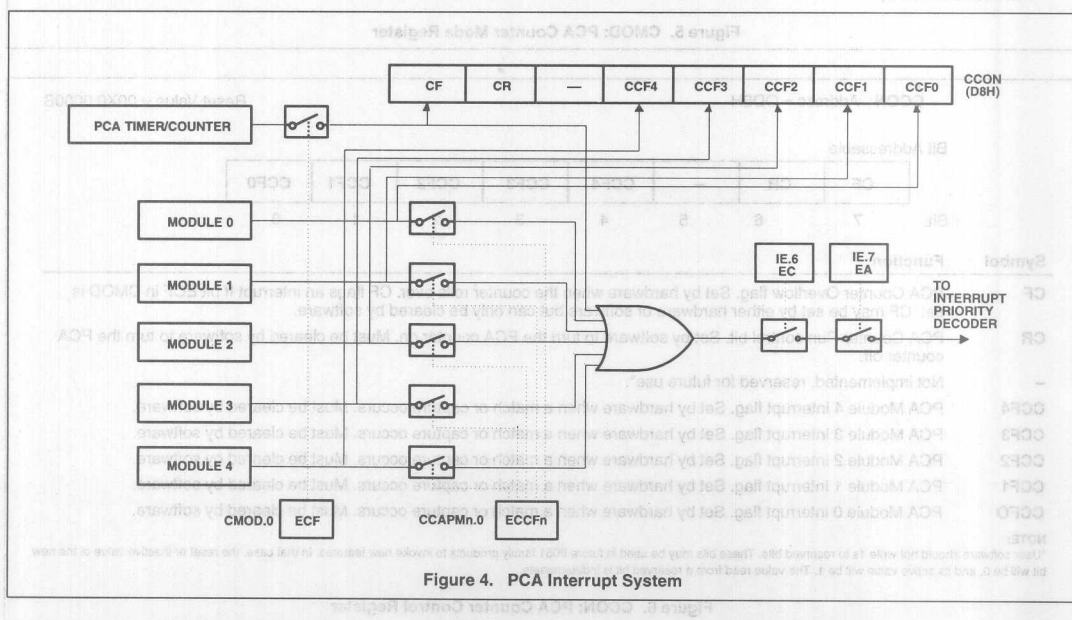


Figure 3. PCA Timer/Counter



**Figure 4. PCA Interrupt System**



## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575

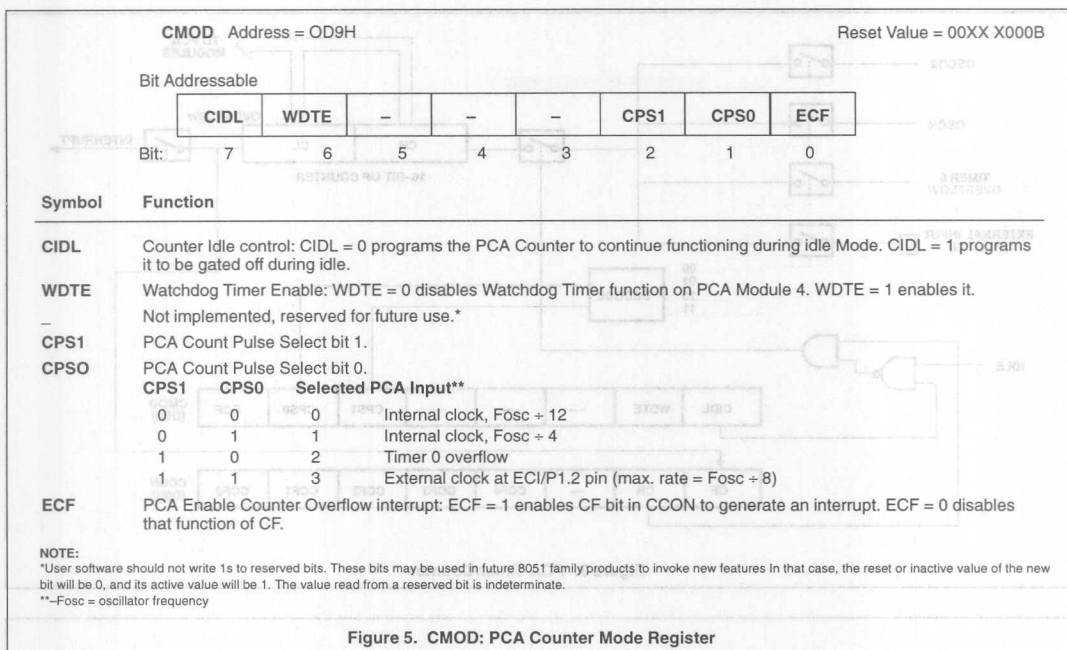


Figure 5. CMOD: PCA Counter Mode Register

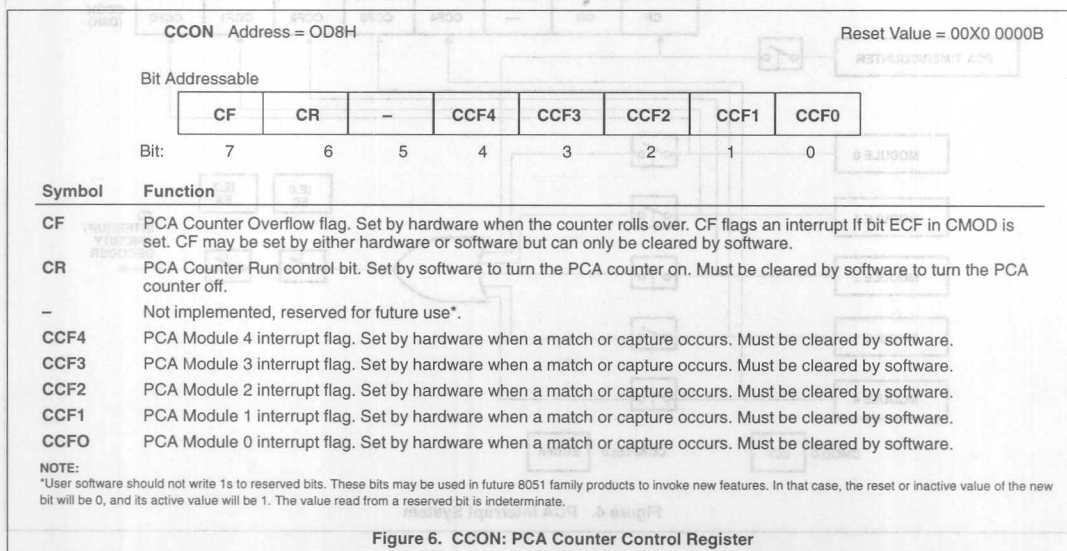
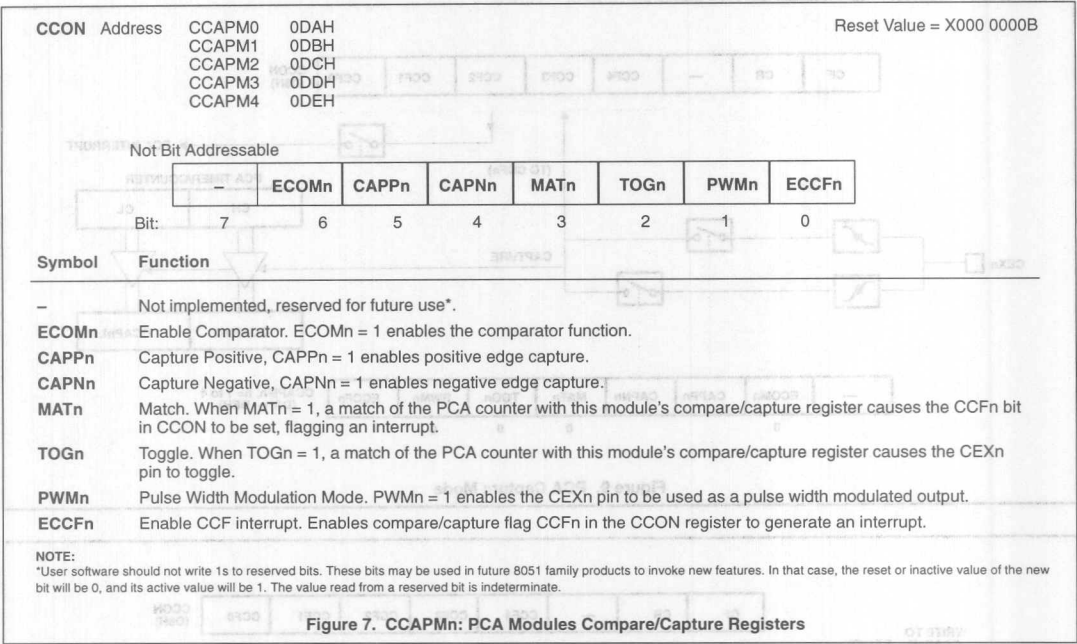


Figure 6. CCON: PCA Counter Control Register

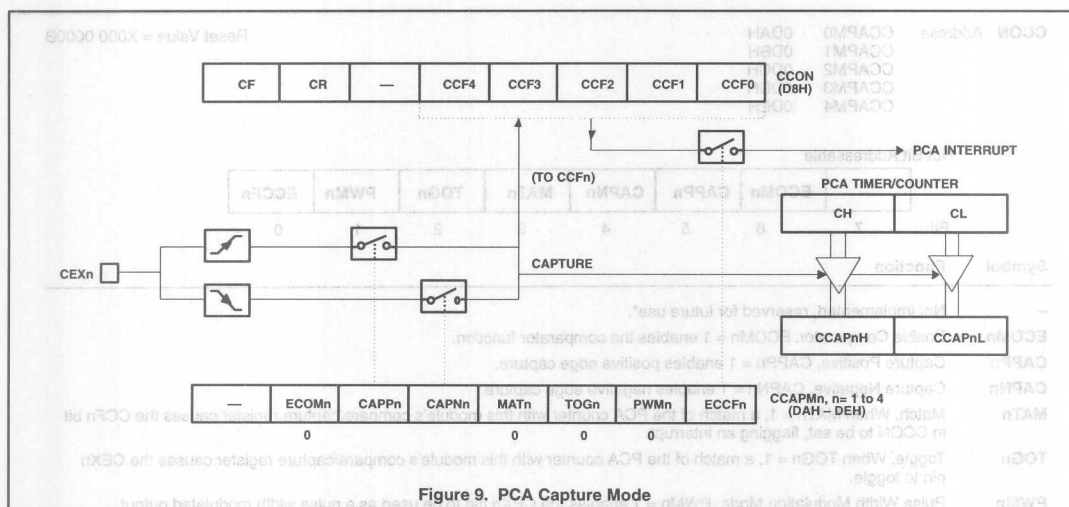
CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575



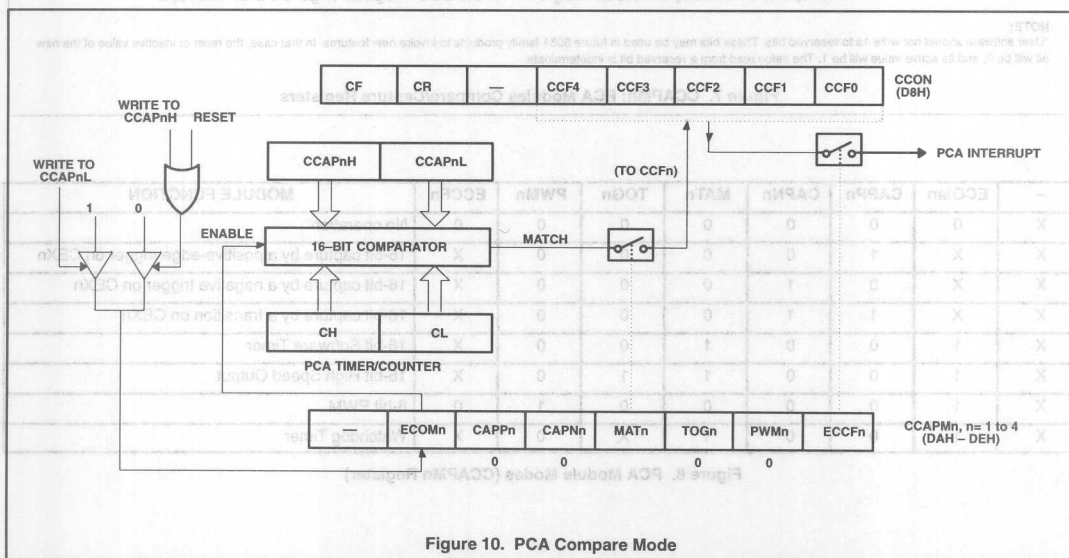
—	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn	MODULE FUNCTION
X	0	0	0	0	0	0	0	No operation
X	X	1	0	0	0	0	X	16-bit capture by a positive-edge trigger on CEXn
X	X	0	1	0	0	0	X	16-bit capture by a negative trigger on CEXn
X	X	1	1	0	0	0	X	16-bit capture by a transition on CEXn
X	1	0	0	1	0	0	X	16-bit Software Timer
X	1	0	0	1	1	0	X	16-bit High Speed Output
X	1	0	0	0	0	1	0	8-bit PWM
X	1	0	0	1	X	0	X	Watchdog Timer

Figure 8. PCA Module Modes (CCAPMn Register)

## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575



### Figure 9. PCA Capture Mode



**Figure 10. PCA Compare Mode**

## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575

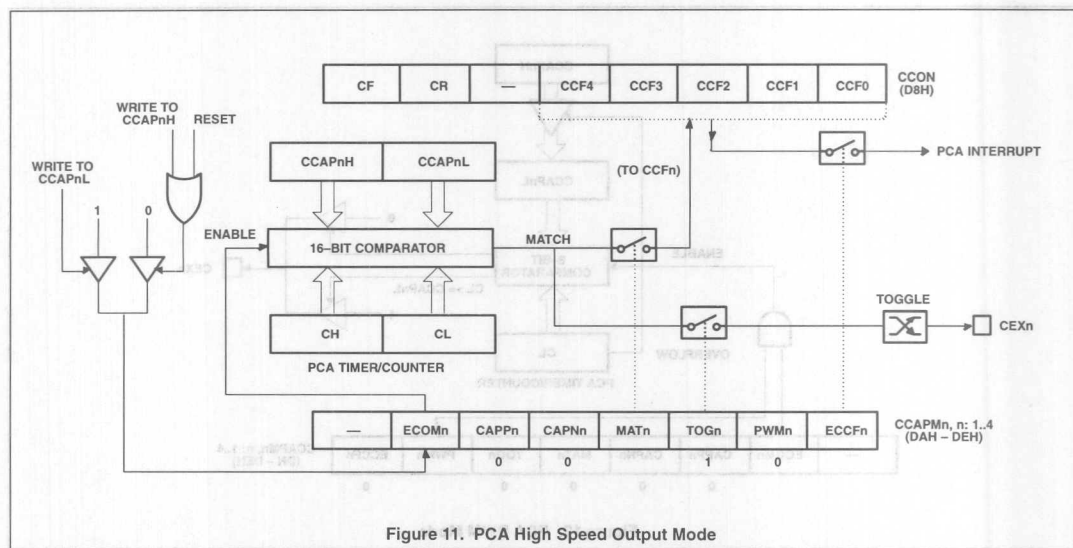


Figure 11. PCA High Speed Output Mode

**Pulse Width Modulator Mode**

All of the PCA modules can be used as PWM outputs. Figure 12 shows the PWM function. The frequency of the output depends on the source for the PCA timer. All of the modules will have the same frequency of output because they all share the PCA timer. The duty cycle of each module is independently variable using the module's capture register CCAPLn. When the value of the PCA CL SFR is less than the value in the module's CCAPLn SFR the output will be low, when it is equal to or greater than the output will be high. When CL overflows from FF to 00, CCAPLn is reloaded with the value in CCAPHn. The PWM and ECOM bits in the module's CCAPMn register must be set to enable the PWM mode.

**WATCHDOG TIMER**

The watchdog timer is not directly loadable by the user. Instead, the value to be loaded into the main timer is held in an autoloader register or is part of the mask ROM programming. In order to cause the main timer to be loaded with the appropriate value, a special sequence of software action must take place. This operation is referred to as feeding the watchdog timer.

To feed the watchdog, two instructions must be sequentially executed successfully. No

intervening instruction fetches are allowed, so interrupts should be disabled before feeding the watchdog. The instructions should move A5H to the WFEED1 register and then 5AH to the WFEED2 register. If WFEED1 is correctly loaded and WFEED2 is not correctly loaded, then an immediate underflow will occur.

The watchdog timer subsystem has two modes of operation. Its principal function is a watchdog timer. In this mode it protects the system from incorrect code execution by causing a system reset when the watchdog timer underflows as a result of a failure of software to feed the timer prior to the timer reaching its terminal count. If the user does not employ the watchdog function, the watchdog subsystem can be used as a timer. In this mode, reaching the terminal count sets a flag. In most other respects, the timer mode possesses the characteristics of the watchdog mode. This is done to protect the integrity of the watchdog function.

The watchdog timer subsystem consists of a prescaler and a main counter. The prescaler has 8 selectable taps off the final stages and the output of a selected tap provides the clock to the main counter. The main counter is the section that is loaded as a result of the software feeding the watchdog and it is the section that causes the system reset

(watchdog mode) or time-out flag to be set (timer mode) if allowed to reach its terminal count.

**Programming the Watchdog Timer**

Both the EPROM and ROM devices have a set of SFRs for holding the watchdog autoloader values and the control bits. The watchdog time-out flag is present in the watchdog control register and operates the same in all versions. In the EPROM device, the watchdog parameters (autoloader value and control) are always taken from the SFRs. In the ROM device, the watchdog parameters can be mask programmed or taken from the SFRs. The selection to take the watchdog parameters from the SFRs or from the mask programmed values is controlled by EA (external access). When EA is high (internal ROM access), the watchdog parameters are taken from the mask programmed values. If the watchdog is mask programmed to the timer mode, then the autoloader values and the pre-scaler taps are taken from the SFRs. When EA is low (external access), the watchdog parameters are taken from the SFRs. The user should be able to leave code in his program which initializes the watchdog SFRs even though he has migrated to the mask ROM part. This allows no code changes from EPROM prototyping to ROM coded production parts.

## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575

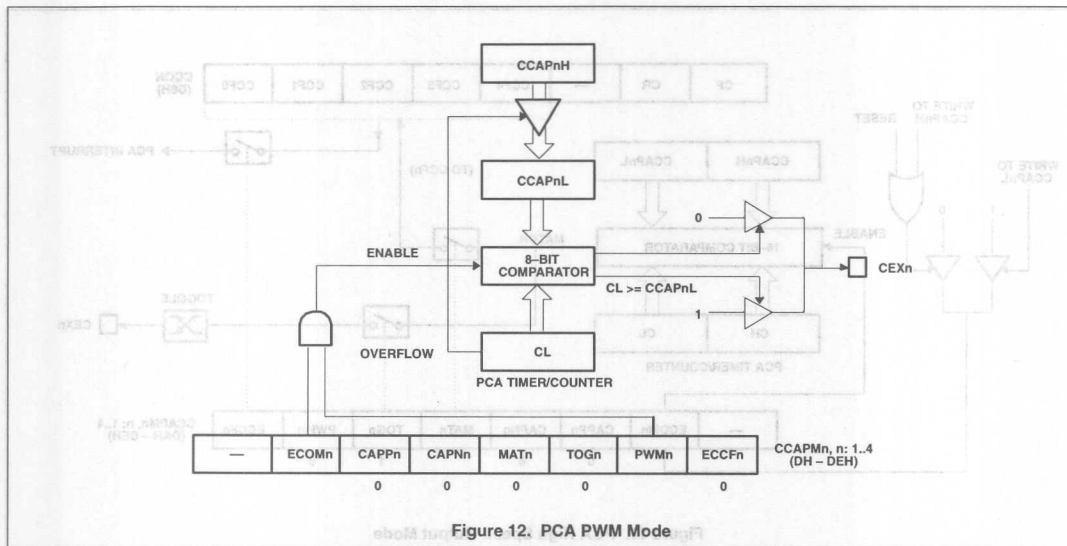


Figure 12. PCA PWM Mode

**Watchdog Detailed Operation****EPROM Device (and ROMless Operation: EA = 0)**

In the ROMless operation (ROM part, EA = 0) and in the EPROM device, the watchdog operates in the following manner (see Figure 14).

Whether the watchdog is in the watchdog or timer mode, when external RESET is applied, the following takes place:

- Watchdog mode bit set to watchdog mode.
- Watchdog run control bit set to ON.
- Autoload register set to 00 (min. count).
- Watchdog time-out flag cleared.
- Prescaler is cleared.
- Prescaler tap set to the highest divide.
- Autoload takes place.

The watchdog can be fed even though it is in the timer mode.

Note that the operational concept is for the watchdog mode of operation, when coming out of a hardware reset, the software should load the autoload registers, set the mode to watchdog, and then feed the watchdog (cause an autoload). The watchdog will now be starting at a known point.

If the watchdog is in the watchdog mode and running and happens to underflow at the time

the external RESET is applied, the watchdog time-out flag will be cleared.

When the watchdog is in the watchdog mode and the watchdog underflows, the following action takes place (see Figure 16):

- Autoload takes place.
- Watchdog time-out flag is set
- Mode bit unchanged.
- Watchdog run bit unchanged.
- Autoload register unchanged.
- Prescaler tap unchanged.
- All other device action same as external reset.

Note that if the watchdog underflows, the program counter will start from 00H as in the case of an external reset. The watchdog time-out flag can be examined to determine if the watchdog has caused the reset condition. The watchdog time-out flag bit can be cleared by software.

When the watchdog is in the timer mode and the timer software underflows, the following action takes place:

- Autoload takes place.
- Watchdog time-out flag is set
- Mode bit unchanged.
- Watchdog run bit unchanged.

- Autoload register unchanged.
- Prescaler tap unchanged.

**Mask ROM Device (EA = 1)**

In the mask ROM device, the watchdog mode bit (WDMOD) is mask programmed and the bit in the watchdog command register is read only and reflects the mask programmed selection. If the mask programmed mode bit selects the timer mode, then the watchdog run bit (WDRUN) operates as described under EPROM Device. If the mask programmed bit selects the watchdog mode, then the watchdog run bit has no effect on the timer operation (see Figure 15).

**Watchdog Function**

The watchdog consists of a programmable prescaler and the main timer. The prescaler derives its clock from the on-chip oscillator. The prescaler consists of a divide by 12 followed by a 13 stage counter with taps from stage 6 through stage 13. This is shown in Figure 17. The tap selection is programmable. The watchdog main counter is a down counter clocked (decremented) each time the programmable prescaler underflows. The watchdog generates an underflow signal (and is autoloading) when the watchdog is at count 0 and the clock to decrement the watchdog occurs. The watchdog is 8 bits long and the autoload value can range from 0 to FFH. (The



## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575

autoload value of 0 is permissible since the prescaler is cleared upon autoload).

This leads to the following user design equations. Definitions:  $t_{OSC}$  is the oscillator period,  $N$  is the selected prescaler tap value,  $W$  is the main counter autoload value,  $t_{MIN}$  is the minimum watchdog time-out value (when the autoload value is 0),  $t_{MAX}$  is the maximum time-out value (when the autoload value is FFH),  $t_D$  is the design time-out value.

$$t_{MIN} = t_{OSC} \times 12 \times 64$$

$$t_{MAX} = t_{MIN} \times 128 \times 256$$

$$t_D = t_{MIN} \times 2^{PRESCALER} \times W + 1$$

(where prescaler = 0, 1, 2, 3, 4, 5, 6, or 7)

Note that the design procedure is anticipated to be as follows. A  $t_{MAX}$  will be chosen either from equipment or operation considerations and will most likely be the next convenient value higher than  $t_D$ . (If the watchdog were inadvertently to start from FFH, an overflow would be guaranteed, barring other anomalies, to occur within  $t_{MAX}$ ). Then the value for the prescaler would be chosen from:

$$\text{prescaler} = \log_2 (t_{MAX} / (t_{OSC} \times 12 \times 256)) - 6$$

This then also fixes  $t_{MIN}$ . An autoload value would then be chosen from:

$$W = t_D / t_{MIN} - 1$$

The software must be written so that a feed operation takes place every  $t_D$  seconds from the last feed operation. Some tradeoffs may need to be made. It is not advisable to include feed operations in minor loops or in subroutines unless the feed operation is a specific subroutine.

#### Watchdog Control Register (WDCON) (Bit Addressable) Address C0

The following bits of this register are read only in the ROM part when EA is high: WDMOD, PRE0, PRE1, and PRE2. That is, the register will reflect the mask programmed values. In the ROM part with EA high, these bits are taken from mask coded bits and are not readable by the program. WDRUN is read only in the ROM part when EA is high and

WDMOD is in the watchdog mode. When WDMOD is in the timer mode, WDRUN functions normally.

The parameters written into WDMOD, PRE0, PRE1, and PRE2 by the program are not applied directly to the watchdog timer subsystem. The watchdog timer subsystem is directly controlled by a second register which stores these bits. The transfer of these bits from the user register (WDMOD) to the second control register takes place when the watchdog is fed. This prevents random code execution from directly foiling the watchdog function. This does not affect the operation where these bits are taken from mask coded values.

The reset values of the WDCON and WDL registers will be such that the timer resets to the watchdog mode with a timeout period of  $12 \times 64 \times 128 \times t_{OSC}$ . The watchdog timer will not generate an interrupt. Additional bits in WDCON are used to disable reset generation by the oscillator fail and low voltage detect circuits. WDCON can be written by software only by executing a valid watchdog feed sequence.

#### WDCON Register Bit Definitions

WDCON.7	PRE2	Prescaler Select 2, reset to 1
WDCON.6	PRE1	Prescaler Select 1, reset to 1
WDCON.5	PRE0	Prescaler Select 0, reset to 1
WDCON.4	LVRE	Low Voltage Reset Enable, reset to 1 (enabled)
WDCON.3	OFRE	Oscillator Fail Reset Enable, reset to 1 (enabled)
WDCON.2	WDRUN	Watchdog Run, reset to 1 (enabled)
WDCON.1	WDTOF	Watchdog Timeout Flag, reset to 0
WDCON.0	WDMOD	Watchdog Mode, reset to 1 (watchdog mode)

#### Enhanced UART

The UART operates in all of the usual modes that are described in the first section of this book for the 80C51. In addition the UART can perform framing error detect by looking for missing stop bits, and automatic address recognition. The 87C575 UART also fully supports multiprocessor communication as does the standard 80C51 UART.

When used for framing error detect the UART looks for missing stop bits in the communication. A missing bit will set the FE bit in the SCON register. The FE bit shares the SCON.7 bit with SM0 and the function of SCON.7 is determined by PCON.6 (SMOD0) (see Figure 19). If SMOD0 is set then SCON.7 functions as FE. SCON.7 functions as SM0 when SMOD0 is cleared. When used as FE SCON.7 can only be cleared by software. Refer to Figure 18.

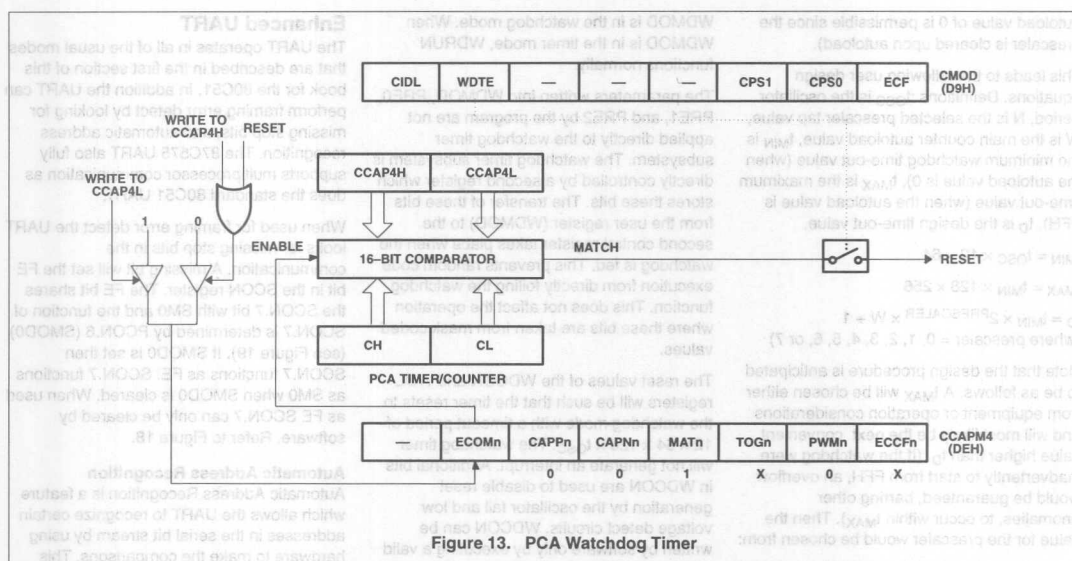
#### Automatic Address Recognition

Automatic Address Recognition is a feature which allows the UART to recognize certain addresses in the serial bit stream by using hardware to make the comparisons. This feature saves a great deal of software overhead by eliminating the need for the software to examine every serial address which passes by the serial port. This feature is enabled by setting the SM2 bit in SCON. In the 9 bit UART modes, mode 2 and mode 3, the Receive Interrupt flag (RI) will be automatically set when the received byte contains either the "Given" address or the "Broadcast" address. The 9 bit mode requires that the 9th information bit is a 1 to indicate that the received information is an address and not data. Automatic address recognition is shown in Figure 20.

The 8 bit mode is called Mode 1. In this mode the RI flag will be set if SM2 is enabled and the information received has a valid stop bit following the 8 address bits and the information is either a Given or Broadcast address.

Mode 0 is the Shift Register mode and SM2 is ignored.

## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575



**Figure 13. PCA Watchdog Timer**

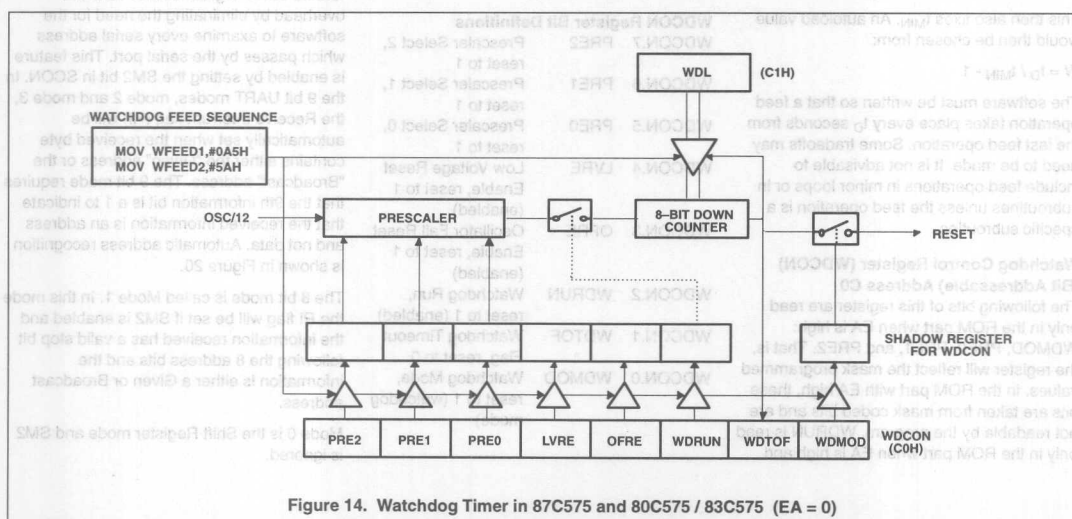


Figure 14. Watchdog Timer in 87C575 and 80C575 / 83C575 (EA = 0)

## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575

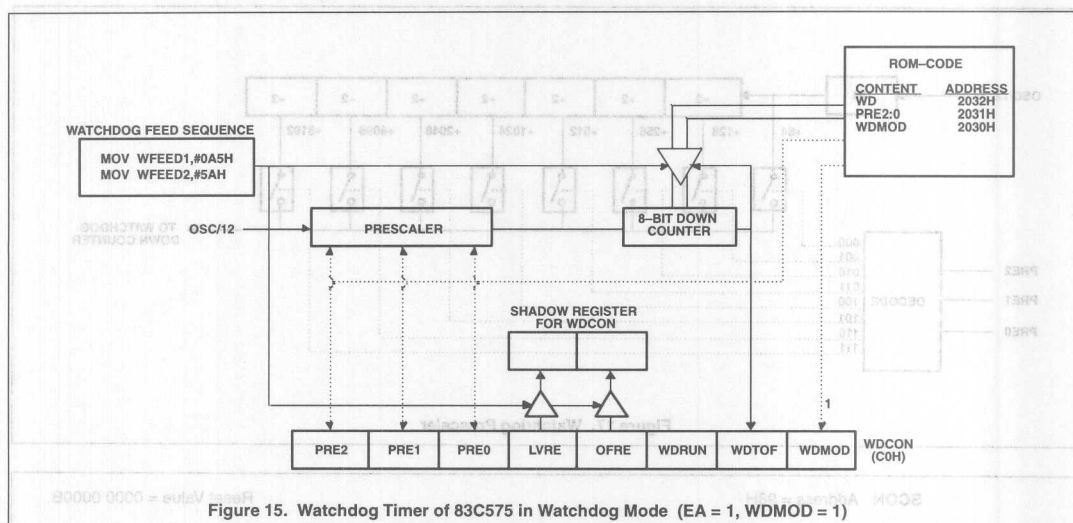


Figure 15. Watchdog Timer of 83C575 in Watchdog Mode (EA = 1, WDMOD = 1)

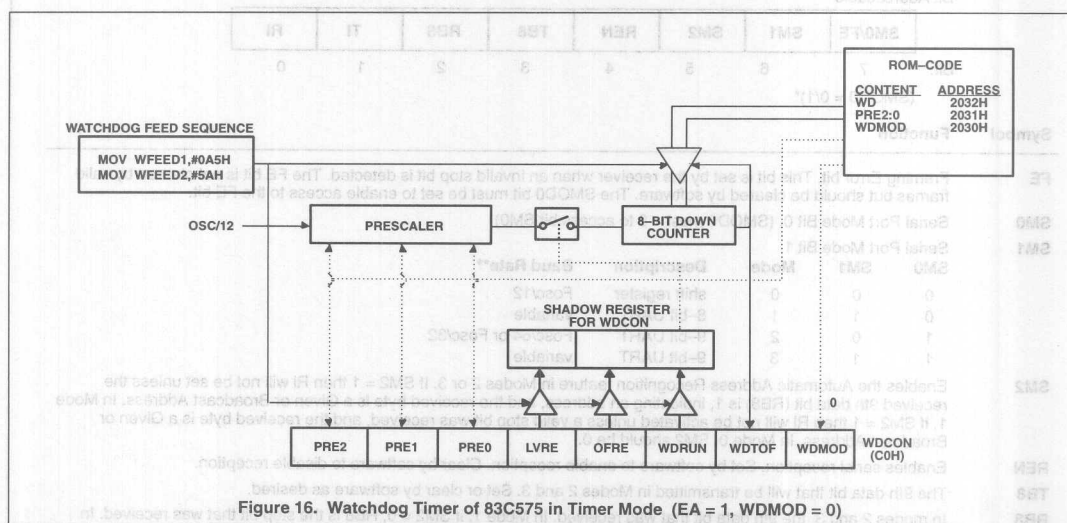


Figure 16. Watchdog Timer of 83C575 in Timer Mode (EA = 1, WDMOD = 0)

CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575

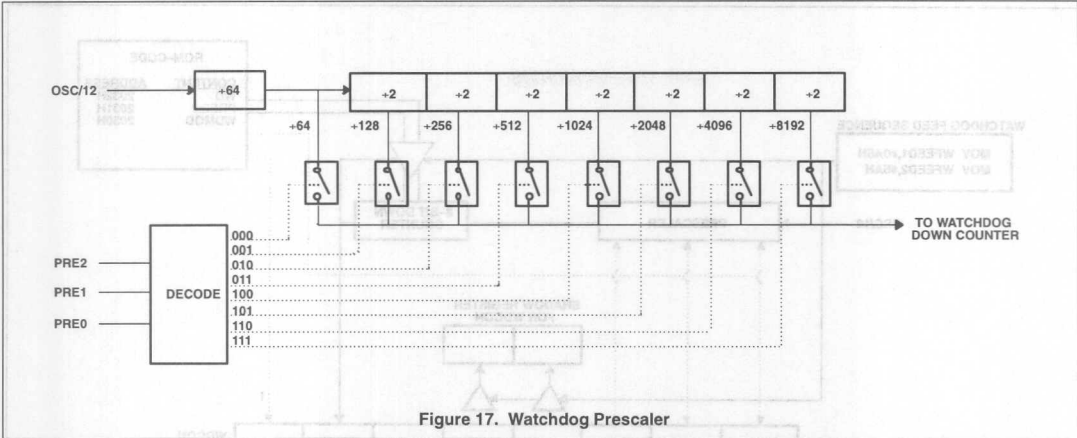


Figure 17. Watchdog Prescaler

SCON Address = 98H

Reset Value = 0000 0000B

Bit Addressable

SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI
--------	-----	-----	-----	-----	-----	----	----

Bit: 7 6 5 4 3 2 1 0

(SMOD0 = 0/1)\*

Symbol Function

FE	Framing Error bit. This bit is set by the receiver when an invalid stop bit is detected. The FE bit is not cleared by valid frames but should be cleared by software. The SMOD0 bit must be set to enable access to the FE bit.														
SM0	Serial Port Mode Bit 0, (SMOD0 must = 0 to access bit SM0)														
SM1	Serial Port Mode Bit 1														
	SM0	SM1	Mode	Description	Baud Rate**										
	0	0	0	shift register	Fosc/12										
	0	1	1	8-bit UART	variable										
	1	0	2	9-bit UART	Fosc/64 or Fosc/32										
	1	1	3	9-bit UART	variable										
SM2	Enables the Automatic Address Recognition feature in Modes 2 or 3. If SM2 = 1 then RI will not be set unless the received 9th data bit (RB8) is 1, indicating an address, and the received byte is a Given or Broadcast Address. In Mode 1, if SM2 = 1 then RI will not be activated unless a valid stop bit was received, and the received byte is a Given or Broadcast Address. In Mode 0, SM2 should be 0.														
REN	Enables serial reception. Set by software to enable reception. Clear by software to disable reception.														
TB8	The 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired.														
RB8	In modes 2 and 3, the 9th data bit that was received. In Mode 1, if SM2 = 0, RB8 is the stop bit that was received. In Mode 0, RB8 is not used.														
TI	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.														
RI	Receive interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM2). Must be cleared by software.														

NOTE:

\*SMOD0 is located at PCON6.

\*\*Fosc = oscillator frequency

Figure 18. SCON: Serial Port Control Register

Figure 18. SCON: Serial Port Control Register

## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575

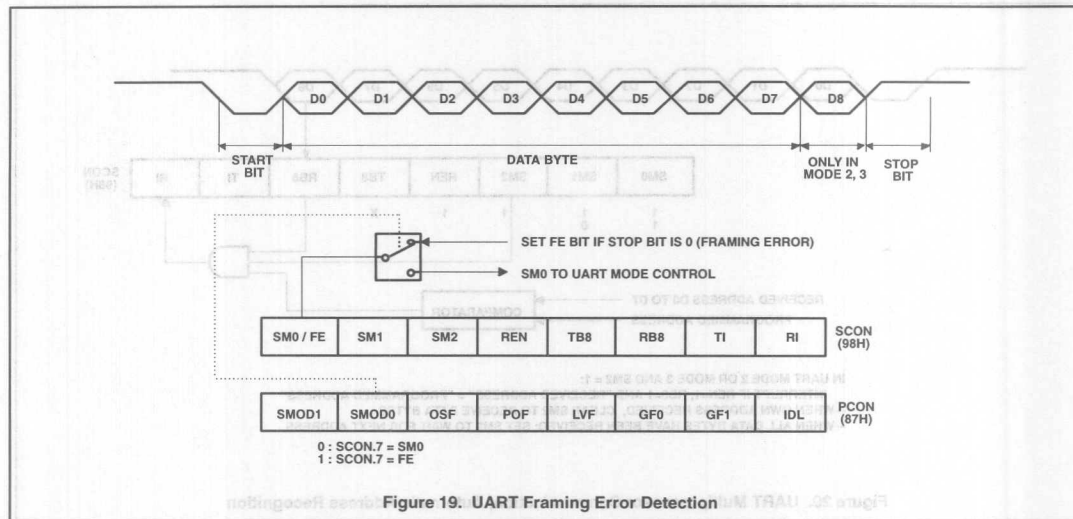


Figure 19. UART Framing Error Detection

Using the Automatic Address Recognition feature allows a master to selectively communicate with one or more slaves by invoking the Given slave address or addresses. All of the slaves may be contacted by using the Broadcast address. Two special Function Registers are used to define the slave's address, SADDR, and the address mask, SADEN. SADEN is used to define which bits in the SADDR are to be used and which bits are "don't care". The SADEN mask can be logically ANDed with the SADDR to create the "Given" address which the master will use for addressing each of the slaves. Use of the Given address allows multiple slaves to be recognized while excluding others. The following examples will help to show the versatility of this scheme:

Slave 0	SADDR =	1100 0000
	SADEN =	1111 1101
	Given =	1100 00X0
Slave 1	SADDR =	1100 0000
	SADEN =	1111 1110
	Given =	1100 00XX

In the above example SADDR is the same and the SADEN data is used to differentiate between the two slaves. Slave 0 requires a 0 in bit 0 and it ignores bit 1. Slave 1 requires a 0 in bit 1 and bit 0 is ignored. A unique address for Slave 0 would be 1100 0010 since slave 1 requires a 0 in bit 1. A unique address for slave 1 would be 1100 0001 since a 1 in bit 0 will exclude slave 0. Both

slaves can be selected at the same time by an address which has bit 0 = 0 (for slave 0) and bit 1 = 0 (for slave 1). Thus, both could be addressed with 1100 0000.

In a more complex system the following could be used to select slaves 1 and 2 while excluding slave 0:

Slave 0	SADDR =	1100 0000
	SADEN =	1111 1001
	Given =	1100 0XX0
Slave 1	SADDR =	1110 0000
	SADEN =	1111 1010
	Given =	1110 0X0X
Slave 2	SADDR =	1110 0000
	SADEN =	1111 1100
	Given =	1110 00XX

In the above example the differentiation among the 3 slaves is in the lower 3 address bits. Slave 0 requires that bit 0 = 0 and it can be uniquely addressed by 1110 0110. Slave 1 requires that bit 1 = 0 and it can be uniquely addressed by 1110 and 0101. Slave 2 requires that bit 2 = 0 and its unique address is 1110 0011. To select Slaves 0 and 1 and exclude Slave 2 use address 1110 0100, since it is necessary to make bit 2 = 1 to exclude slave 2.

The Broadcast Address for each slave is created by taking the logical OR of SADDR and SADEN. Zeros in this result are treated

as don't-cares. In most cases, interpreting the don't-cares as ones, the broadcast address will be FF hexadecimal.

Upon reset SADDR (SFR address 0A9H) and SADEN (SFR address 0B9H) are loaded with 0s. This produces a given address of all "don't cares" as well as a Broadcast address of all "don't cares". This effectively disables the Automatic Addressing mode and allows the microcontroller to use standard 80C51 type UART drivers which do not make use of this feature.

### Analog Comparators

Four analog comparators are provided on chip. Three comparators have a common negative reference CMPR- and independent positive inputs CMP1+, CMP2+, CMP3+ on port 3. The fourth comparator has independent positive and negative inputs CMP0+ and CMP0- on port 1. The CMP register contains an output and enable bit for each comparator. The CMP register is bit addressable and is located at SFR address E8H. Figure 21 shows the connection of the comparators.

Pullups at the comparator input pins will be disabled by hardware when the comparator is enabled. In addition, to make inputs high impedance, the corresponding port SFR bits must be set by software to disable the pulldowns.



## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575

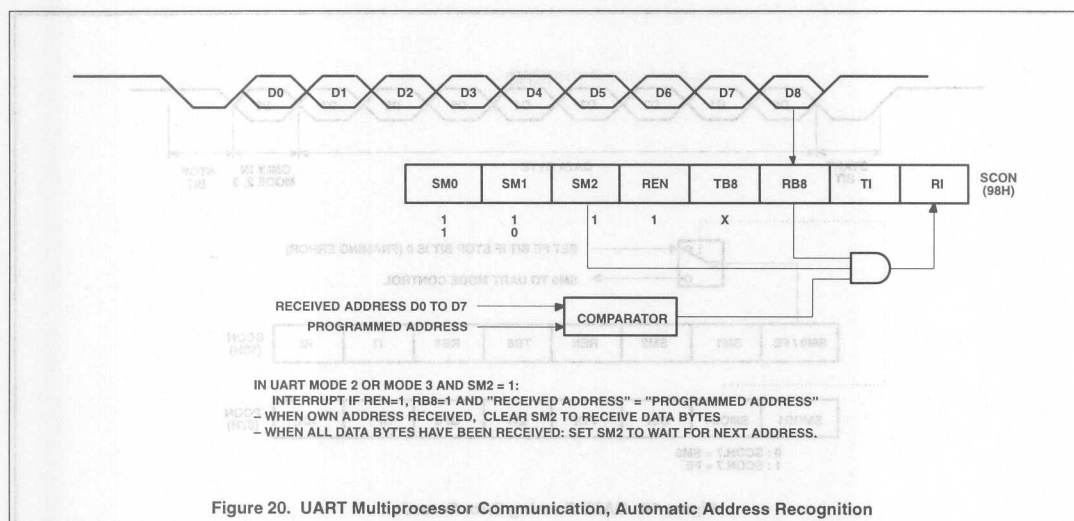


Figure 20. UART Multiprocessor Communication, Automatic Address Recognition

**CMP Register Bit Definitions**

- CMP.7 enable comparator 3, disable pullups at P3.4, P3.7  
 CMP.6 enable comparator 2, disable pullups at P3.4, P3.6  
 CMP.5 enable comparator 1, disable pullups at P3.4, P3.5  
 CMP.4 enable comparator 0, disable pullups at P1.0, P1.1  
 CMP.3 comparator 3 output (read only)  
 CMP.2 comparator 2 output (read only)  
 CMP.1 comparator 1 output (read only)  
 CMP.0 comparator 0 output (read only)

All comparators are disabled automatically in power down mode, in idle mode unused comparators should be disabled by software to save power. A comparator can generate an interrupt that will terminate idle mode when used to drive a PCA capture input.

The CMPE register contains bits to enable each comparator to drive external output pins or internal PCA capture inputs. Pullups at the output pins are disabled by hardware when the external comparator output is enabled. The comparator output is wire-ORed with the corresponding port SFR bit, so the SFR bit must also be set by software to enable the output.

**CMPE Register Bit Definitions**

- CMPE.7 enables comparator 3 to drive CEX3  
 CMPE.6 enables comparator 2 to drive CEX2  
 CMPE.5 enables comparator 1 to drive CEX1  
 CMPE.4 enables comparator 0 to drive CEX0  
 CMPE.3 enables comparator 3 output on P1.6 (open drain)  
 CMPE.2 enables comparator 2 output on P1.5 (open drain)  
 CMPE.1 enables comparator 1 output on P1.4 (open drain)  
 CMPE.0 enables comparator 0 output on P1.3 (open drain)

When 1s are written to CMPE bits 7-4, the comparator outputs will drive the corresponding capture input. When 1s are written to CMPE bits 3-0 the comparator output will also drive the corresponding port 1 pin. If the comparator's enabled to drive the capture input but not the port pin, then the port pin can be used for general purpose I/O. When a comparator output is enabled, pullups at the output pin are disabled and the output becomes open drain.

There are two special function registers associated with the comparators. They are CMP which contains the comparator enables and a bit that can be read by software to determine the state of each comparator's output, and CMPE which controls whether the output from each comparator drives the

associated output pin or a capture input associated with one of the PCA modules.

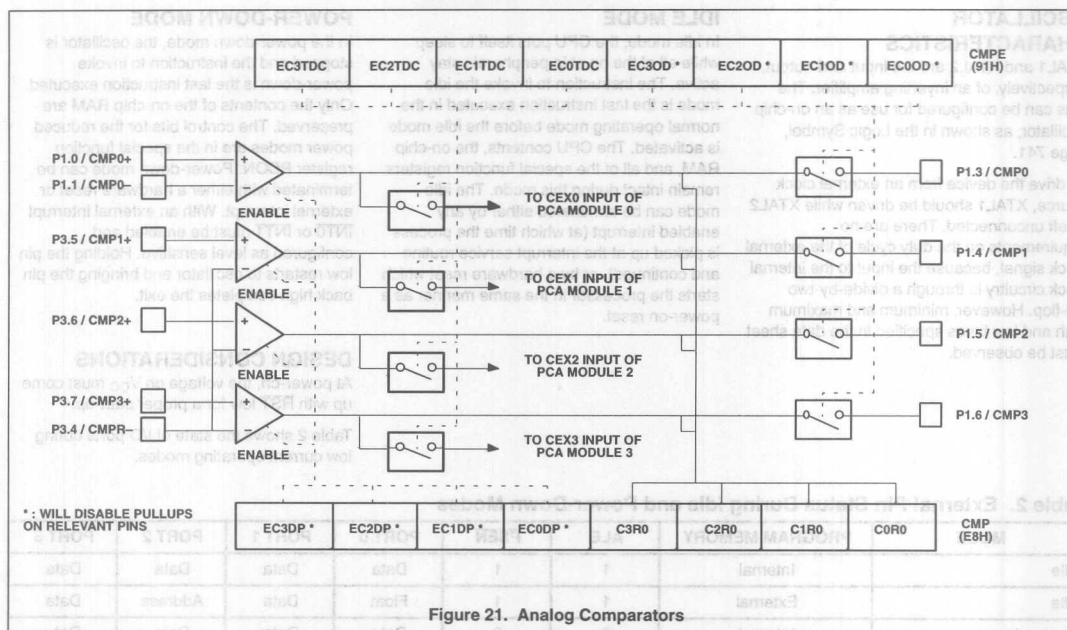
The CMP registers bits 0-3 can be read by software to determine the state of the output of each comparator. To do this the associated comparator must be enabled but the output in port 1 can be disabled. This allows easy polling of the comparator output value without the need to use up a port pin.

The CMPE register allows the comparator to drive the associated PCA module capture input, so that on compare a capture can be generated in the PCA. Bits 0-3 of this register enable the comparator output to drive the associated port 1 output circuitry. Used as a comparator output this circuitry is open drain. To enable the comparator output to drive to port 1, the corresponding port bit must also be set to disable the pulldown. If the comparator is not enabled to drive the port 1 circuitry, the associated port 1 pin can be used for other I/O. This includes when a comparator is enabled to drive the capture input to a PCA module.

**Reduced EMI Mode**

There are two bits in the AUXR register that can be set to reduce the internal clock drive and disable the ALE output. AO (AUXR.0) when set turns off the ALE output. LO (AUXR.1) when set reduces the drive of the internal clock circuitry. Both bits are cleared on Reset. With LO set the 87C575 will still operate at 12MHz, but will have reduced EMI in the range above 100MHz.

## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575



## 8XC575 Reduced EMI Mode

## AUXR (0X8E)

—	—	—	—	—	—	LO	AO
---	---	---	---	---	---	----	----

AO: Turns off ALE output.

LO: Reduces drive of internal clock circuitry. 8XC575 spec'd to 12MHz when LO set.

## INTERNAL RESET

Internal resets generated by the power on, low voltage, and oscillator fail detect circuits are self timed to guarantee proper initialization of the 8XC575. Reset will be held approximately 24 oscillator periods after normal conditions are detected by all enabled detect circuits. Internal resets do not drive RST but will cause missing pulses on ALE.

## Interrupt Enable (IE) Register

EA	IE.7	enable all interrupts
EC	IE.6	enable PCA interrupt
ET2	IE.5	enable Timer 2 interrupt
ES	IE.4	enable Serial I/O interrupt
ET1	IE.3	enable Timer 1 interrupt
EX1	IE.2	enable External interrupt 1
ET0	IE.1	enable Timer 0 interrupt
EX0	IE.0	enable External interrupt 0

## Interrupt Priority (IP) Register

	IP.7	reserved
PPC	IP.6	PCA interrupt priority
PT2	IP.5	Timer 2 interrupt priority
PS	IP.4	Serial I/O interrupt priority
PT1	IP.3	Timer 1 interrupt priority
PX1	IP.2	External interrupt 1 priority
PT0	IP.1	Timer 0 interrupt priority
PX0	IP.0	External interrupt 0 priority

Priority	Source	Flag	Vector
1	INT0	IE0	03H highest priority
2	Timer 0	TF0	0BH
3	INT1	IE1	13H
4	Timer 1	TF1	1BH
5	PCA	CF,CCFn	33H
6	Serial I/O	RI,TI	23H
7	Timer 2	TF2/EXF2	2BH lowest priority

## Power Control (PCON) Register

SMOD1	PCON.7	double baud rate bit
SMOD0	PCON.6	SCON.7 access control
OSF	PCON.5	oscillator fail flag
POF	PCON.4	power off flag
LVF	PCON.3	low voltage flag
GF0	PCON.2	general purpose flag
PD	PCON.1	power down mode bit
IDL	PCON.0	idle mode bit

## Auxiliary Register Bit Definitions (AUXR = 8EH)

AO AUXR.0 ALE Off,  
when set turns off ALE

LO AUXR.1 Low Speed,  
reduces internal clock drive

## Port 2 Pullup Disable Register (P2OD = 0A1H)

Port 2 pullups can be disabled by writing ones to P2OD. Each bit in P2OD controls the corresponding bit in P2. P2OD resets to all zeros enabling Port 2 pullups. Writing one to a P2OD bit disables pullups at the corresponding port 2 bit making the output open drain.

CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator, as shown in the Logic Symbol, page 741.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

IDLE MODE

In idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

POWER-DOWN MODE

In the power-down mode, the oscillator is stopped and the instruction to invoke power-down is the last instruction executed. Only the contents of the on-chip RAM are preserved. The control bits for the reduced power modes are in the special function register PCON. Power-down mode can be terminated with either a hardware reset or external interrupt. With an external interrupt INT0 or INT1 must be enabled and configured as level sensitive. Holding the pin low restarts to oscillator and bringing the pin back high completes the exit.

DESIGN CONSIDERATIONS

At power-on, the voltage on VCC must come up with RST low for a proper start-up.

Table 2 shows the state of I/O ports during low current operating modes.

Table 2. External Pin Status During Idle and Power-Down Modes

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575

ROM CODE SUBMISSION

When submitting ROM code for the 83C575, the following must be specified:

- 1. 8k byte user ROM data
- 2. 32 byte ROM encryption key
- 3. ROM security bits
- 4. The watchdog timer parameters.

ADDRESS	CONTENT	BIT(S)	COMMENT
0000H to 1FFFFH	DATA	7:0	User ROM Data
2000H to 201FH	KEY	7:0	ROM Encryption Key FFH = no encryption
2020H	Reserved	2	Must = 1
	Security Bit 2	1	0 = enable, 1 = disable
	Security Bit 1	0	0 = enable, 1 = disable
2030H	Reserved	7:0	Must = FFH
2031H	Reserved	7:0	Must = FFH
2032H	WDL <sup>1</sup>	7:0	Watchdog reload value (see specification)
2033H	WDCON <sup>1</sup>	7:5	PRE2:0
2033H	WDCON <sup>1</sup>	4	LVRE
2033H	WDCON <sup>1</sup>	3	OFRE
2033H	WDCON <sup>1</sup>	2	WDF=0, not ROM coded
2033H	WDCON <sup>1</sup>	1	WDRUN=0, not ROM coded
2033H	WDCON <sup>1</sup>	0	WDMOD

NOTES:

- 1. See Watchdog Timer Specification for definition of WDL and WDCON bits.

**Security Bit 1:** When programmed, this bit has two effects on masked ROM parts:

- 1. External MOVX is disabled, and
- 2. EA# is latched on Reset.

**Security Bit 2:** When programmed, this bit inhibits Verify User ROM.

ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>

PARAMETER	RATING	UNIT
Operating temperature under bias	-55 to +125	°C
Storage temperature range	-65 to +150	°C
Voltage on EA/V <sub>PP</sub> pin to V <sub>SS</sub>	0 to +13.0	V
Voltage on any other pin to V <sub>SS</sub>	-0.5 to +6.5	V
Maximum I <sub>OL</sub> per I/O pin	15	mA
Power dissipation (based on package heat transfer limitations, not device power consumption)	1.5	W

NOTES:

- 1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
- 2. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
- 3. Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V<sub>SS</sub> unless otherwise noted.

## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575

## DC ELECTRICAL CHARACTERISTICS

$T_{amb} = 0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$  and  $-40^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 20\%$ ,  $V_{SS} = 0\text{V}$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP <sup>1</sup>	MAX	
$V_{IL}$	Input low voltage (Ports 0, 2, 3, except 3.2, 3.3)		-0.5		$0.5V_{CC}-0.6$	V
$V_{IL1}$	Input low voltage (Ports 1, 3.2, 3.3)		-0.5		$0.65V_{CC}-0.5$	V
$V_{IL2}$	Input low voltage (EA)		0		$0.2V_{CC}-0.45$	V
$V_{IL3}$	Input low voltage (XTAL1, RST)		-0.5		$0.2V_{CC}-0.1$	V
$V_{IH}$	Input high voltage (Ports 0, 2, 3, except 3.2, 3.3)		$0.5V_{CC}+0.8$		$V_{CC}+0.5$	V
$V_{IH1}$	Input high voltage (Ports 1, 3.2, 3.3)		$0.8V_{CC}+0.3$		$V_{CC}+0.5$	V
$V_{IH2}$	Input high voltage (EA)		$0.2V_{CC}+0.9$		$V_{CC}+0.5$	V
$V_{IH3}$	Input high voltage (XTAL1, RST)		$0.7V_{CC}$		$V_{CC}+0.5$	V
HYS	Hysteresis (Ports 0, 2, 3, except 3.2, 3.3)		200			mV
HYS1	Hysteresis (Ports 1, 3.2, 3.3)		50			mV
$V_{OL}$	Output voltage low (Ports 1, 2, 3, except 3.1)	$I_{OL} = 1.6\text{mA}$			0.45	V
$V_{OL1}$	Output voltage low (Ports 0, ALE, PSEN)	$I_{OL} = 3.2\text{mA}$			0.45	V
$V_{OL2}$	Output voltage low P3.1 with bit cleared P3.1 with bit set	$I_{OL} = 10.0\text{mA}$ $I_{OL} = 1.6\text{mA}$			0.50	V
					0.45	V
$V_{OH}$	Output voltage high (Ports 1, 2, 3, except P3.1)	$I_{OH} = -30\mu\text{A}$ $I_{OH} = -10\mu\text{A}$	$V_{CC}-0.7$ $V_{CC}-0.3$			V
						V
$V_{OH1}$	Output voltage high (Port 0 in external bus mode, ALE, PSEN)	$I_{OH} = -3.2\text{mA}$ $I_{OH} = -200\mu\text{A}$	$V_{CC}-0.7$ $V_{CC}-0.3$			V
						V
$V_{OH2}$	Output voltage high P3.1 with bit cleared P3.1 with bit set	$I_{OH} = -10.0\text{mA}$ $I_{OH} = -1.6\text{mA}$	$V_{CC}-1.5$ $V_{CC}-1.5$			V
						V
$V_{IO}$	Offset voltage comparator inputs		-35		+35	mV
$V_{CR}$	Common mode range comparator inputs		0		$V_{CC}$	V
$I_{IL}$	Logical 0 input current (Ports 1, 2, 3, except 3.1)	$V_{IN} = 0.45\text{V}$			-75	$\mu\text{A}$
$I_{TL}$	Logical 1-to-0 transition current (Ports 2, 3, except 3.1, 3.2, 3.3) <sup>4</sup>	See Note 4			-600	$\mu\text{A}$
$I_{TL1}$	Logical 1-to-0 transition current (Ports 1, 3.2, 3.3)	See Note 4			-450	$\mu\text{A}$
$I_{L1}$	Input leakage current (Port 0, Port2 in open drain mode) <sup>9</sup>	$0.45 < V_{IN} < V_{CC}$	2		40	$\mu\text{A}$
$I_{L2}$	Input leakage current (EA, P3.1)	$0.45 < V_{IN} < V_{CC}$	-10		+10	$\mu\text{A}$
$I_{LC}$	Input leakage current comparator inputs	$0 < V_{IN} < V_{CC}$	-1.0		+1.0	$\mu\text{A}$
$I_{CC}$	Power supply current: <sup>7</sup> Active mode @ 16MHz <sup>5</sup> Idle mode @ 16MHz Power-down mode	See note 6		20	30	mA
				8	12	mA
				5	75	$\mu\text{A}$
$R_{RST}$	Internal reset pull-up resistor	$V_{IN} = 0\text{V}$	50		200	k $\Omega$
$V_{LOW}$	Low $V_{CC}$ detect voltage		4.0		4.45	V
$C_{IO}$	Pin capacitance <sup>10</sup>	$f = 1\text{MHz}$			10	pF

NOTES: (SEE NEXT PAGE)



## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575

## NOTES TO THE DC ELECTRICAL CHARACTERISTICS TABLE:

- Typical ratings are not guaranteed. The values listed are at room temperature, 5V.
- Capacitive loading on ports 0 and 2 may cause spurious noise to be superimposed on the  $V_{OL}$ s of ALE and ports 1 and 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100pF), the noise pulse on the ALE pin may exceed 0.8V. In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.  $I_{OL}$  can exceed these conditions provided that no single output sinks more than 5mA and no more than two outputs exceed the test conditions.
- Capacitive loading on ports 0 and 2 may cause the  $V_{OH}$  on ALE and  $PSEN$  to momentarily fall below the 0.9V<sub>CC</sub> specification when the address bits are stabilizing.
- Pins of ports 1, 2 and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when  $V_{IN}$  is between  $V_{IH}$  and  $V_{IL}$ .
- $I_{CCMAX}$  at other frequencies can be determined from Figure 29.
- See Figures 30 through 33 for  $I_{CC}$  test conditions.
- Load capacitance for port 0, ALE, and  $PSEN$  = 100pF, load capacitance for all other outputs = 80pF.
- Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:
 

Maximum $I_{OL}$ per port pin:	10mA
Maximum $I_{OL}$ per 8-bit port:	26mA
Maximum total $I_{OL}$ for all outputs:	71mA

 If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
- Specification applies to Port 2 when P2OD bit is set.
- 15pF MAX for the EA/V<sub>PP</sub> and P0.0 pins.

Symbol	Parameter	Min	Max	Unit
$V_{CC}$	Supply voltage	4.5	5.5	V
$V_{DD}$	Data memory supply voltage	4.5	5.5	V
$V_{EE}$	EEPROM supply voltage	4.5	5.5	V
$V_{IL}$	Input low voltage	0	0.8	V
$V_{IH}$	Input high voltage	2.0	5.5	V
$V_{OL}$	Output low voltage	0	0.4	V
$V_{OH}$	Output high voltage	2.4	5.5	V
$I_{CC}$	Supply current	0.5	1.0	mA
$I_{CCMAX}$	Maximum supply current	10	15	mA
$I_{OL}$	Output low current	0	10	mA
$I_{OH}$	Output high current	0	10	mA
$I_{P0}$	Port 0 current	0	10	mA
$I_{P1}$	Port 1 current	0	10	mA
$I_{P2}$	Port 2 current	0	10	mA
$I_{P3}$	Port 3 current	0	10	mA
$I_{P4}$	Port 4 current	0	10	mA
$I_{P5}$	Port 5 current	0	10	mA
$I_{P6}$	Port 6 current	0	10	mA
$I_{P7}$	Port 7 current	0	10	mA
$I_{P8}$	Port 8 current	0	10	mA
$I_{P9}$	Port 9 current	0	10	mA
$I_{P10}$	Port 10 current	0	10	mA
$I_{P11}$	Port 11 current	0	10	mA
$I_{P12}$	Port 12 current	0	10	mA
$I_{P13}$	Port 13 current	0	10	mA
$I_{P14}$	Port 14 current	0	10	mA
$I_{P15}$	Port 15 current	0	10	mA
$I_{P16}$	Port 16 current	0	10	mA
$I_{P17}$	Port 17 current	0	10	mA
$I_{P18}$	Port 18 current	0	10	mA
$I_{P19}$	Port 19 current	0	10	mA
$I_{P20}$	Port 20 current	0	10	mA
$I_{P21}$	Port 21 current	0	10	mA
$I_{P22}$	Port 22 current	0	10	mA
$I_{P23}$	Port 23 current	0	10	mA
$I_{P24}$	Port 24 current	0	10	mA
$I_{P25}$	Port 25 current	0	10	mA
$I_{P26}$	Port 26 current	0	10	mA
$I_{P27}$	Port 27 current	0	10	mA
$I_{P28}$	Port 28 current	0	10	mA
$I_{P29}$	Port 29 current	0	10	mA
$I_{P30}$	Port 30 current	0	10	mA
$I_{P31}$	Port 31 current	0	10	mA
$I_{P32}$	Port 32 current	0	10	mA
$I_{P33}$	Port 33 current	0	10	mA
$I_{P34}$	Port 34 current	0	10	mA
$I_{P35}$	Port 35 current	0	10	mA
$I_{P36}$	Port 36 current	0	10	mA
$I_{P37}$	Port 37 current	0	10	mA
$I_{P38}$	Port 38 current	0	10	mA
$I_{P39}$	Port 39 current	0	10	mA
$I_{P40}$	Port 40 current	0	10	mA
$I_{P41}$	Port 41 current	0	10	mA
$I_{P42}$	Port 42 current	0	10	mA
$I_{P43}$	Port 43 current	0	10	mA
$I_{P44}$	Port 44 current	0	10	mA
$I_{P45}$	Port 45 current	0	10	mA
$I_{P46}$	Port 46 current	0	10	mA
$I_{P47}$	Port 47 current	0	10	mA
$I_{P48}$	Port 48 current	0	10	mA
$I_{P49}$	Port 49 current	0	10	mA
$I_{P50}$	Port 50 current	0	10	mA
$I_{P51}$	Port 51 current	0	10	mA
$I_{P52}$	Port 52 current	0	10	mA
$I_{P53}$	Port 53 current	0	10	mA
$I_{P54}$	Port 54 current	0	10	mA
$I_{P55}$	Port 55 current	0	10	mA
$I_{P56}$	Port 56 current	0	10	mA
$I_{P57}$	Port 57 current	0	10	mA
$I_{P58}$	Port 58 current	0	10	mA
$I_{P59}$	Port 59 current	0	10	mA
$I_{P60}$	Port 60 current	0	10	mA
$I_{P61}$	Port 61 current	0	10	mA
$I_{P62}$	Port 62 current	0	10	mA
$I_{P63}$	Port 63 current	0	10	mA
$I_{P64}$	Port 64 current	0	10	mA
$I_{P65}$	Port 65 current	0	10	mA
$I_{P66}$	Port 66 current	0	10	mA
$I_{P67}$	Port 67 current	0	10	mA
$I_{P68}$	Port 68 current	0	10	mA
$I_{P69}$	Port 69 current	0	10	mA
$I_{P70}$	Port 70 current	0	10	mA
$I_{P71}$	Port 71 current	0	10	mA
$I_{P72}$	Port 72 current	0	10	mA
$I_{P73}$	Port 73 current	0	10	mA
$I_{P74}$	Port 74 current	0	10	mA
$I_{P75}$	Port 75 current	0	10	mA
$I_{P76}$	Port 76 current	0	10	mA
$I_{P77}$	Port 77 current	0	10	mA
$I_{P78}$	Port 78 current	0	10	mA
$I_{P79}$	Port 79 current	0	10	mA
$I_{P80}$	Port 80 current	0	10	mA
$I_{P81}$	Port 81 current	0	10	mA
$I_{P82}$	Port 82 current	0	10	mA
$I_{P83}$	Port 83 current	0	10	mA
$I_{P84}$	Port 84 current	0	10	mA
$I_{P85}$	Port 85 current	0	10	mA
$I_{P86}$	Port 86 current	0	10	mA
$I_{P87}$	Port 87 current	0	10	mA
$I_{P88}$	Port 88 current	0	10	mA
$I_{P89}$	Port 89 current	0	10	mA
$I_{P90}$	Port 90 current	0	10	mA
$I_{P91}$	Port 91 current	0	10	mA
$I_{P92}$	Port 92 current	0	10	mA
$I_{P93}$	Port 93 current	0	10	mA
$I_{P94}$	Port 94 current	0	10	mA
$I_{P95}$	Port 95 current	0	10	mA
$I_{P96}$	Port 96 current	0	10	mA
$I_{P97}$	Port 97 current	0	10	mA
$I_{P98}$	Port 98 current	0	10	mA
$I_{P99}$	Port 99 current	0	10	mA
$I_{P100}$	Port 100 current	0	10	mA
$I_{P101}$	Port 101 current	0	10	mA
$I_{P102}$	Port 102 current	0	10	mA
$I_{P103}$	Port 103 current	0	10	mA
$I_{P104}$	Port 104 current	0	10	mA
$I_{P105}$	Port 105 current	0	10	mA
$I_{P106}$	Port 106 current	0	10	mA
$I_{P107}$	Port 107 current	0	10	mA
$I_{P108}$	Port 108 current	0	10	mA
$I_{P109}$	Port 109 current	0	10	mA
$I_{P110}$	Port 110 current	0	10	mA
$I_{P111}$	Port 111 current	0	10	mA
$I_{P112}$	Port 112 current	0	10	mA
$I_{P113}$	Port 113 current	0	10	mA
$I_{P114}$	Port 114 current	0	10	mA
$I_{P115}$	Port 115 current	0	10	mA
$I_{P116}$	Port 116 current	0	10	mA
$I_{P117}$	Port 117 current	0	10	mA
$I_{P118}$	Port 118 current	0	10	mA
$I_{P119}$	Port 119 current	0	10	mA
$I_{P120}$	Port 120 current	0	10	mA
$I_{P121}$	Port 121 current	0	10	mA
$I_{P122}$	Port 122 current	0	10	mA
$I_{P123}$	Port 123 current	0	10	mA
$I_{P124}$	Port 124 current	0	10	mA
$I_{P125}$	Port 125 current	0	10	mA
$I_{P126}$	Port 126 current	0	10	mA
$I_{P127}$	Port 127 current	0	10	mA
$I_{P128}$	Port 128 current	0	10	mA
$I_{P129}$	Port 129 current	0	10	mA
$I_{P130}$	Port 130 current	0	10	mA
$I_{P131}$	Port 131 current	0	10	mA
$I_{P132}$	Port 132 current	0	10	mA
$I_{P133}$	Port 133 current	0	10	mA
$I_{P134}$	Port 134 current	0	10	mA
$I_{P135}$	Port 135 current	0	10	mA
$I_{P136}$	Port 136 current	0	10	mA
$I_{P137}$	Port 137 current	0	10	mA
$I_{P138}$	Port 138 current	0	10	mA
$I_{P139}$	Port 139 current	0	10	mA
$I_{P140}$	Port 140 current	0	10	mA
$I_{P141}$	Port 141 current	0	10	mA
$I_{P142}$	Port 142 current	0	10	mA
$I_{P143}$	Port 143 current	0	10	mA
$I_{P144}$	Port 144 current	0	10	mA
$I_{P145}$	Port 145 current	0	10	mA
$I_{P146}$	Port 146 current	0	10	mA
$I_{P147}$	Port 147 current	0	10	mA
$I_{P148}$	Port 148 current	0	10	mA
$I_{P149}$	Port 149 current	0	10	mA
$I_{P150}$	Port 150 current	0	10	mA
$I_{P151}$	Port 151 current	0	10	mA
$I_{P152}$	Port 152 current	0	10	mA
$I_{P153}$	Port 153 current	0	10	mA
$I_{P154}$	Port 154 current	0	10	mA
$I_{P155}$	Port 155 current	0	10	mA
$I_{P156}$	Port 156 current	0	10	mA
$I_{P157}$	Port 157 current	0	10	mA
$I_{P158}$	Port 158 current	0	10	mA
$I_{P159}$	Port 159 current	0	10	mA
$I_{P160}$	Port 160 current	0	10	mA
$I_{P161}$	Port 161 current	0	10	mA
$I_{P162}$	Port 162 current	0	10	mA
$I_{P163}$	Port 163 current	0	10	mA
$I_{P164}$	Port 164 current	0	10	mA
$I_{P165}$	Port 165 current	0	10	mA
$I_{P166}$	Port 166 current	0	10	mA
$I_{P167}$	Port 167 current	0	10	mA
$I_{P168}$	Port 168 current	0	10	mA
$I_{P169}$	Port 169 current	0	10	mA
$I_{P170}$	Port 170 current	0	10	mA
$I_{P171}$	Port 171 current	0	10	mA
$I_{P172}$	Port 172 current	0	10	mA
$I_{P173}$	Port 173 current	0	10	mA
$I_{P174}$	Port 174 current	0	10	mA
$I_{P175}$	Port 175 current	0	10	mA
$I_{P176}$	Port 176 current	0	10	mA
$I_{P177}$	Port 177 current	0	10	mA
$I_{P178}$	Port 178 current	0	10	mA
$I_{P179}$	Port 179 current	0	10	mA
$I_{P180}$	Port 180 current	0	10	mA
$I_{P181}$	Port 181 current	0	10	mA
$I_{P182}$	Port 182 current	0	10	mA
$I_{P183}$	Port 183 current	0	10	mA
$I_{P184}$	Port 184 current	0	10	mA
$I_{P185}$	Port 185 current	0	10	mA
$I_{P186}$	Port 186 current	0	10	mA
$I_{P187}$	Port 187 current	0	10	mA
$I_{P188}$	Port 188 current	0	10	mA
$I_{P189}$	Port 189 current	0	10	mA
$I_{P190}$	Port 190 current	0	10	mA
$I_{P191}$	Port 191 current	0	10	mA
$I_{P192}$	Port 192 current	0	10	mA
$I_{P193}$	Port 193 current	0	10	mA
$I_{P194}$	Port 194 current	0	10	mA
$I_{P195}$	Port 195 current	0	10	mA
$I_{P196}$	Port 196 current	0	10	mA
$I_{P197}$	Port 197 current	0	10	mA
$I_{P198}$	Port 198 current	0	10	mA
$I_{P199}$	Port 199 current	0	10	mA
$I_{P200}$	Port 200 current	0	10	mA
$I_{P201}$	Port 201 current	0	10	mA
$I_{P202}$	Port 202 current	0	10	mA
$I_{P203}$	Port 203 current	0	10	mA
$I_{P204}$	Port 204 current	0	10	mA
$I_{P205}$	Port 205 current	0	10	mA
$I_{P206}$	Port 206 current	0	10	mA
$I_{P207}$	Port 207 current	0	10	mA
$I_{P208}$	Port 208 current	0	10	mA
$I_{P209}$	Port 209 current	0	10	mA
$I_{P210}$	Port 210 current	0	10	mA
$I_{P211}$	Port 211 current	0	10	mA
$I_{P212}$	Port 212 current	0	10	mA
$I_{P213}$	Port 213 current	0	10	mA
$I_{P214}$	Port 214 current	0	10	mA
$I_{P215}$	Port 215 current	0	10	mA
$I_{P216}$	Port 216 current	0	10	mA
$I_{P217}$	Port 217 current	0	10	mA
$I_{P218}$	Port 218 current	0	10	mA
$I_{P219}$	Port 219 current	0	10	mA
$I_{P220}$	Port 220 current	0	10	mA
$I_{P221}$	Port 221 current	0	10	mA
$I_{P222}$	Port 222 current	0	10	mA

## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575

## AC ELECTRICAL CHARACTERISTICS

T<sub>amb</sub> = 0°C to +70°C and -40°C to +125°C, V<sub>CC</sub> = 5V ±20%, V<sub>SS</sub> = 0V<sup>1,2</sup>

SYMBOL	FIGURE	PARAMETER	VARIABLE CLOCK		UNIT
			MIN	MAX	
1/t <sub>CLCL</sub>	22	Oscillator frequency: <b>Speed Versions</b> 8XC575 E	6	16	MHz
OSCF		Oscillator fail detect frequency	0.6	5.5	MHz
TR		Comparator response time		10	µs
t <sub>LHLL</sub>	22	ALE pulse width	2t <sub>CLCL</sub> -40		ns
t <sub>AVLL</sub>	22	Address valid to ALE low	t <sub>CLCL</sub> -25		ns
t <sub>LLAX</sub>	22	Address hold after ALE low	t <sub>CLCL</sub> -25		ns
t <sub>LLIV</sub>	22	ALE low to valid instruction in		4t <sub>CLCL</sub> -75	ns
t <sub>LLPL</sub>	22	ALE low to PSEN low	t <sub>CLCL</sub> -25		ns
t <sub>PLPH</sub>	22	PSEN pulse width	3t <sub>CLCL</sub> -45		ns
t <sub>PLIV</sub>	22	PSEN low to valid instruction in		3t <sub>CLCL</sub> -70	ns
t <sub>PIXI</sub>	22	Input instruction hold after PSEN	0		ns
t <sub>PIXI</sub>	22	Input instruction float after PSEN		t <sub>CLCL</sub> -25	ns
t <sub>AVIV</sub>	22	Address to valid instruction in		5t <sub>CLCL</sub> -85	ns
t <sub>PLAZ</sub>	22	PSEN low to address float		10	ns
<b>Data Memory</b>					
t <sub>RLRH</sub>	23, 24	RD pulse width	6t <sub>CLCL</sub> -100		ns
t <sub>WLWH</sub>	23, 24	WR pulse width	6t <sub>CLCL</sub> -100		ns
t <sub>RLDV</sub>	23, 24	RD low to valid data in		5t <sub>CLCL</sub> -110	ns
t <sub>RHDX</sub>	23, 24	Data hold after RD	0		ns
t <sub>RHDZ</sub>	23, 24	Data float after RD		2t <sub>CLCL</sub> -28	ns
t <sub>LLDV</sub>	23, 24	ALE low to valid data in		8t <sub>CLCL</sub> -150	ns
t <sub>AVDV</sub>	23, 24	Address to valid data in		9t <sub>CLCL</sub> -165	ns
t <sub>LLWL</sub>	23, 24	ALE low to RD or WR low	3t <sub>CLCL</sub> -50	3t <sub>CLCL</sub> +50	ns
t <sub>AVWL</sub>	23, 24	Address valid to WR low or RD low	4t <sub>CLCL</sub> -75		ns
t <sub>QVWX</sub>	23, 24	Data valid to WR transition	t <sub>CLCL</sub> -30		ns
t <sub>WHQX</sub>	23, 24	Data hold after WR	t <sub>CLCL</sub> -25		ns
t <sub>RLAZ</sub>	23, 24	RD low to address float		0	ns
t <sub>WHLH</sub>	23, 24	RD or WR high to ALE high	t <sub>CLCL</sub> -25	t <sub>CLCL</sub> +25	ns
<b>External Clock</b>					
t <sub>CHCX</sub>	26	High time	12		ns
t <sub>CLCX</sub>	26	Low time	12		ns
t <sub>CLCH</sub>	26	Rise time		20	ns
t <sub>CHCL</sub>	26	Fall time		20	ns
<b>Shift Register</b>					
t <sub>XLXL</sub>	25	Serial port clock cycle time	12t <sub>CLCL</sub>		ns
t <sub>QVXH</sub>	25	Output data setup to clock rising edge	10t <sub>CLCL</sub> -133		ns
t <sub>XHQX</sub>	25	Output data hold after clock rising edge	2t <sub>CLCL</sub> -60		ns
t <sub>XHDX</sub>	25	Input data hold after clock rising edge	0		ns
t <sub>XHDV</sub>	25	Clock rising edge to input data valid		10t <sub>CLCL</sub> -133	ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- Interfacing the 80C32/52 to devices with float times up to 45ns is permitted. This limited bus contention will not cause damage to Port 0 drivers.

## 80C575/83C575/87C575

### EXPLANATION OF THE AC SYMBOLS

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

A – Address

C - Clock

D – Input data

H – Logic level high

I – Instruction (program memory contents)

L – Logic level low, or ALE

P - PSEN

Q – Output data

R -  $\overline{RD}$  signal

t - Time

V - Valid

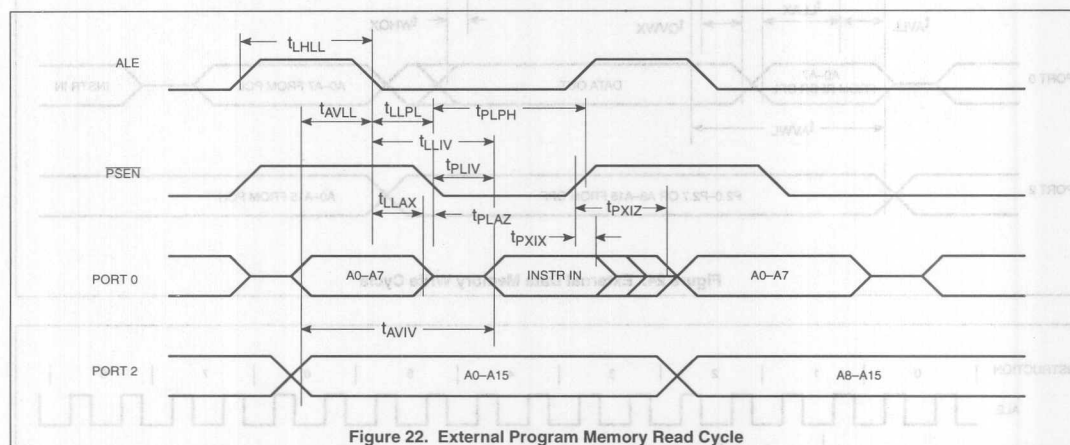
W-  $\overline{WR}$  signal

X – No longer a valid logic level

Z - Float

**Examples:**  $t_{AVLL}$  = Time for address valid to ALE low.

$t_{LLPL}$  = Time for ALE low to  $\overline{PSEN}$  low.



**Figure 22. External Program Memory Read Cycle**

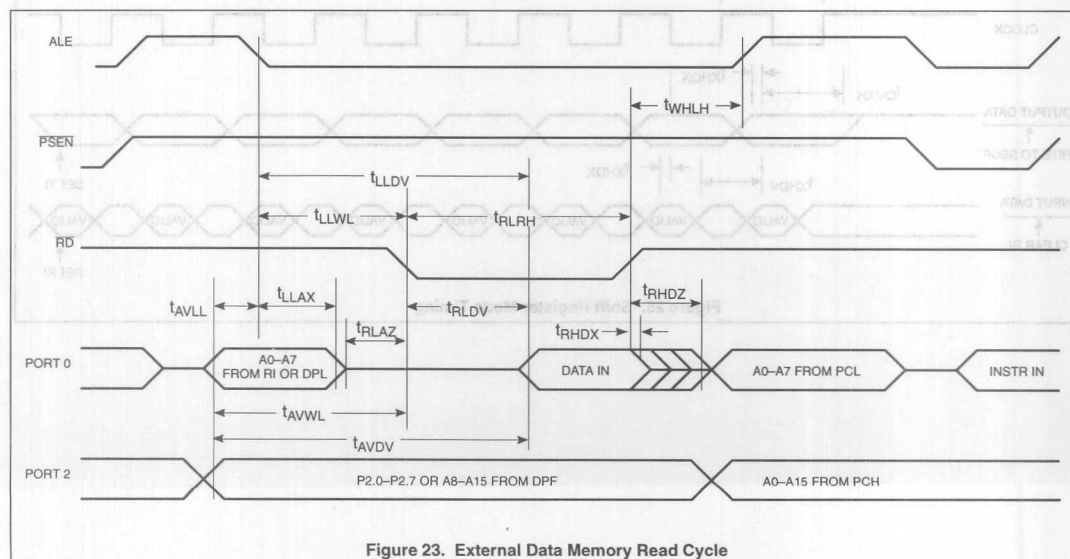


Figure 23. External Data Memory Read Cycle

## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575

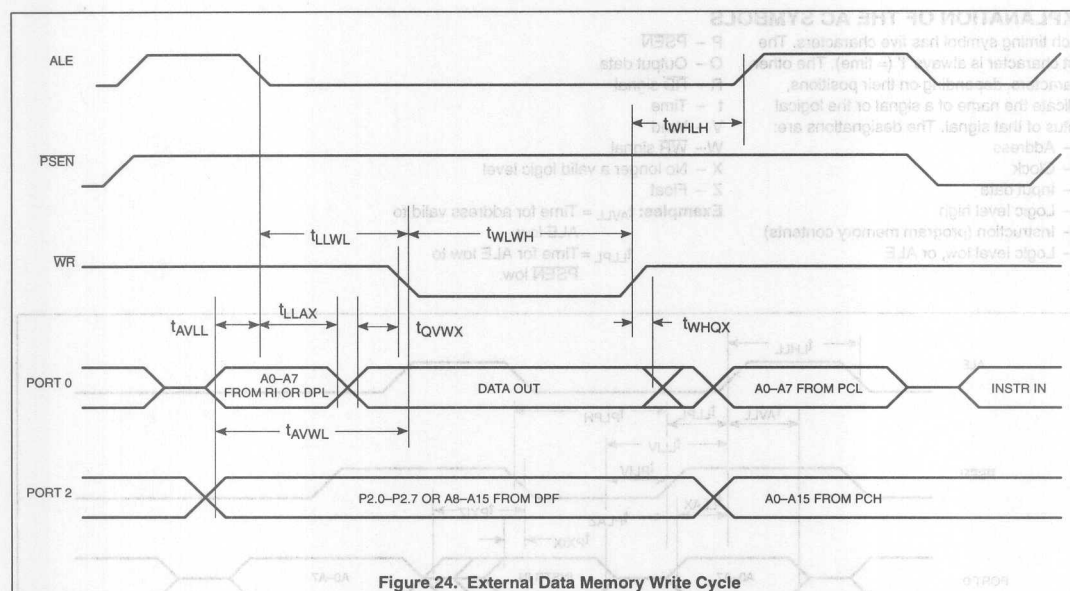


Figure 24. External Data Memory Write Cycle

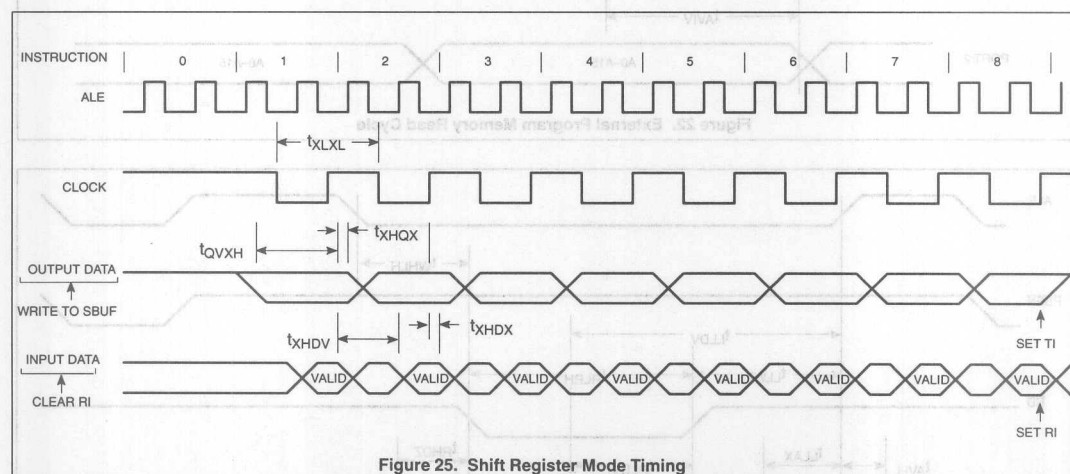
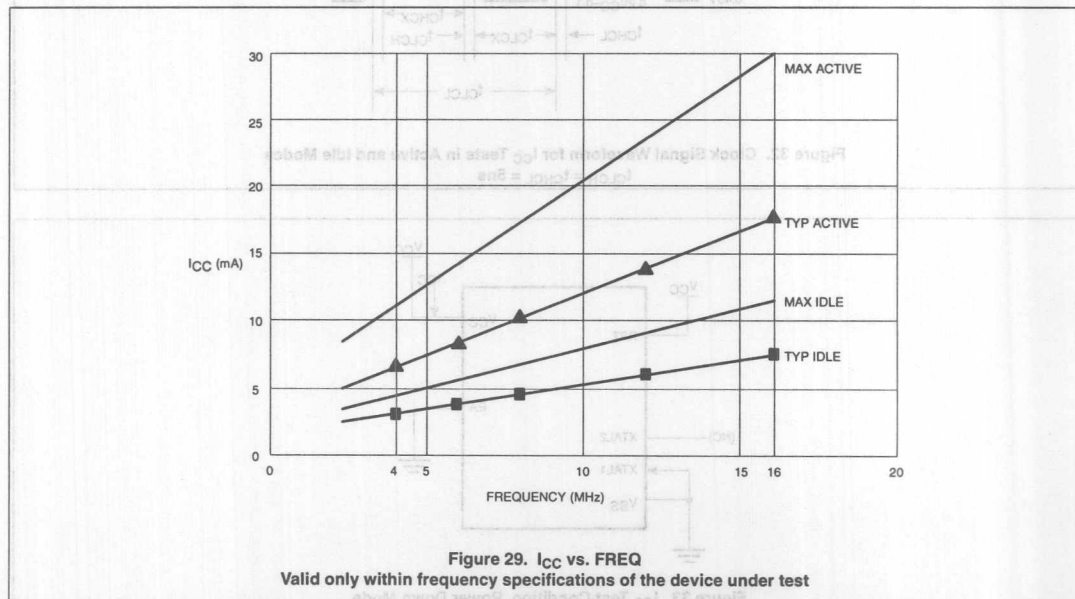
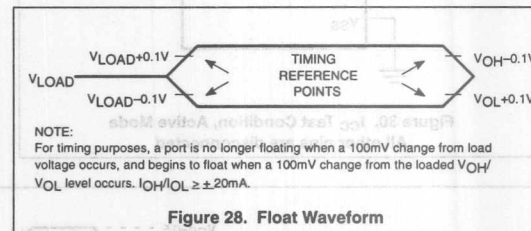
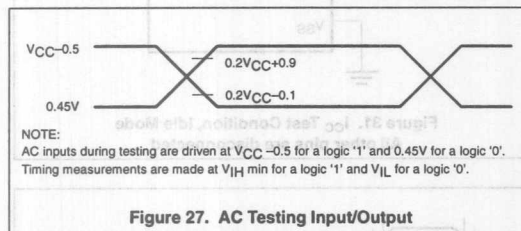
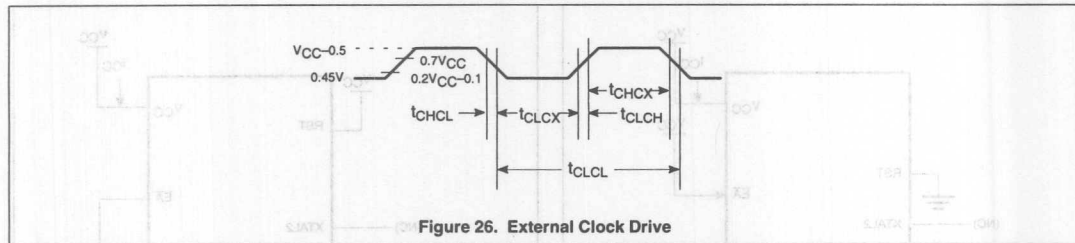
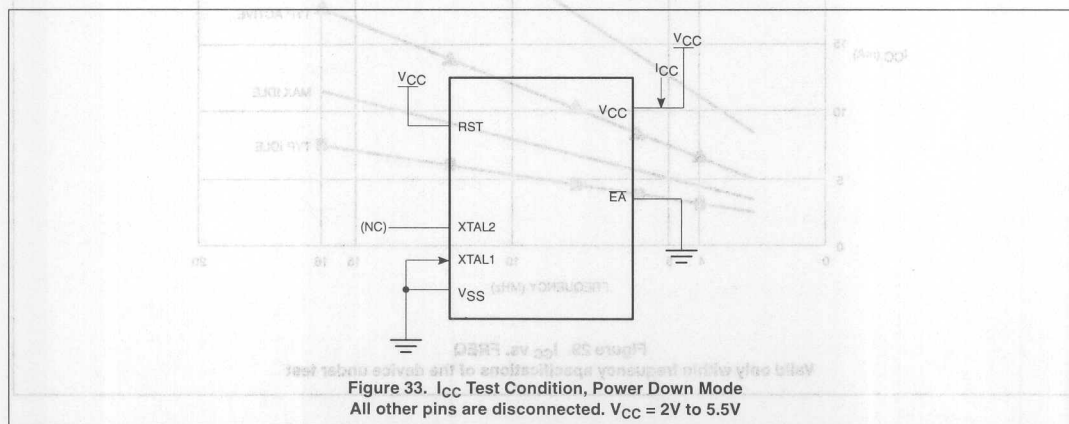
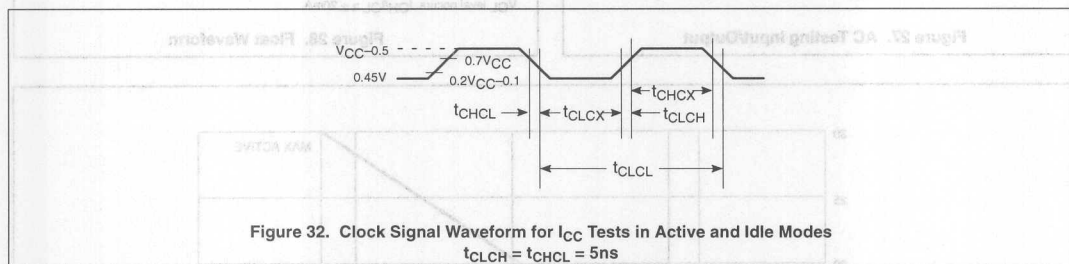
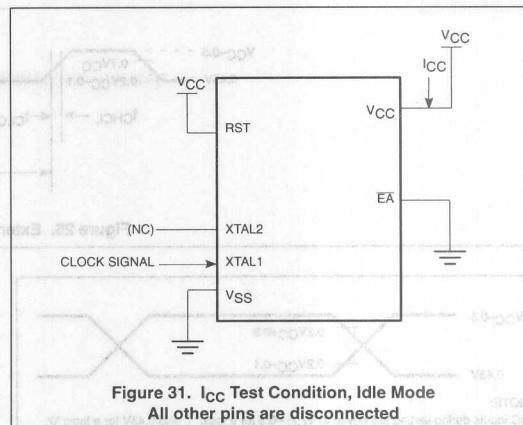
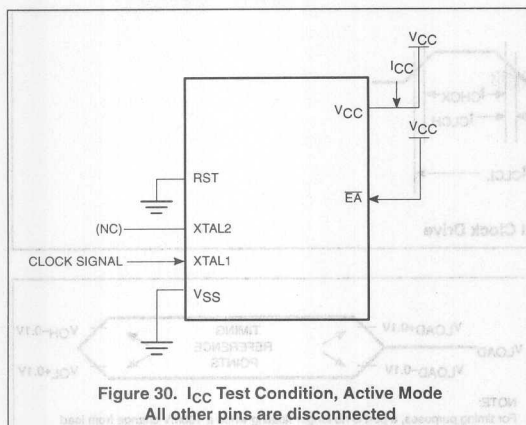


Figure 25. Shift Register Mode Timing

## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575







## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575

**EPROM CHARACTERISTICS**

To put the 87C575 in the EPROM programming mode, PSEN must be held high during power up, then driven low with reset active. The 87C575 is programmed by using a modified Quick-Pulse Programming™ algorithm. It differs from older methods in the value used for V<sub>PP</sub> (programming supply voltage) and in the width and number of the ALE/PROG pulses.

The 87C575 contains two signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes identify the device as an 87C575 manufactured by Philips.

Table 3 shows the logic levels for reading the signature byte, and for programming the program memory, the encryption table, and the security bits. The circuit configuration and waveforms for quick-pulse programming are shown in Figures 34 and 35. Figure 36 shows the circuit configuration for normal program memory verification.

**Quick-Pulse Programming**

The setup for microcontroller quick-pulse programming is shown in Figure 34. Note that the 87C575 is running with a 4 to 6MHz oscillator. The reason the oscillator needs to be running is that the device is executing internal address and program data transfers.

The address of the EPROM location to be programmed is applied to ports 1 and 2, as shown in Figure 34. The code byte to be programmed into that location is applied to port 0. RST, PSEN and pins of ports 2 and 3 specified in Table 3 are held at the 'Program Code Data' levels indicated in Table 3. The ALE/PROG is pulsed low 25 times as shown in Figure 35.

To program the encryption table, repeat the 25 pulse programming sequence for addresses 0 through 1FH, using the 'Pgm Encryption Table' levels. Do not forget that after the encryption table is programmed, verification cycles will produce only encrypted data.

To program the security bits, repeat the 25 pulse programming sequence using the 'Pgm Security Bit' levels. After one security bit is programmed, further programming of the code memory and encryption table is disabled. However, the other security bit can still be programmed.

Note that the EA/V<sub>PP</sub> pin must not be allowed to go above the maximum specified V<sub>PP</sub> level for any amount of time. Even a narrow glitch above that voltage can cause permanent damage to the device. The V<sub>PP</sub> source should be well regulated and free of glitches and overshoot.

**Program Verification**

If security bit 2 has not been programmed, the on-chip program memory can be read out for program verification. The address of the program memory locations to be read is applied to ports 1 and 2 as shown in Figure 36. The other pins are held at the 'Verify Code Data' levels indicated in Table 3. The contents of the address location will be emitted on port 0. External pull-ups are required on port 0 for this operation.

If the encryption table has been programmed, the data presented at port 0 will be the exclusive NOR of the program byte with one of the encryption bytes. The user will have to know the encryption table contents in order to correctly decode the verification data. The encryption table itself cannot be read out.

**Reading the Signature Bytes**

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. The values are:

(030H) = 15H indicates manufactured by Philips  
(B0H) = 97H indicates 87C575

**Program/Verify Algorithms**

Any algorithm in agreement with the conditions listed in Table 3, and which satisfies the timing specifications, is suitable.

**Erase Characteristics**

Erase of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent erasure. **For this and secondary effects, it is recommended that an opaque label be placed over the window.** For elevated temperature or environments where solvents are being used, apply Kapton tape Fluorglas part number 2345-5, or equivalent.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 angstroms) to an integrated dose of at least 15W-s/cm<sup>2</sup>. Exposing the EPROM to an ultraviolet lamp of 12,000µW/cm<sup>2</sup> rating for 20 to 39 minutes, at a distance of about 1 inch, should be sufficient.

Erase leaves the array in an all 1s state.

**Table 3. EPROM Programming Modes**

MODE	RST	PSEN	ALE/PROG	EA/V <sub>PP</sub>	P2.7	P2.6	P3.7	P3.6
Read signature	0	0	1	1	0	0	0	0
Program code data	0	0	0*	V <sub>PP</sub>	1	0	1	1
Verify code data	0	0	1	1	0	0	1	1
Pgm encryption table	0	0	0*	V <sub>PP</sub>	1	0	1	0
Pgm security bit 1	0	0	0*	V <sub>PP</sub>	1	1	1	1
Pgm security bit 2	0	0	0*	V <sub>PP</sub>	1	1	0	0

**NOTES:**

1. '0' = Valid low for that pin, '1' = valid high for that pin.

2. V<sub>PP</sub> = 12.75V ±0.25V.

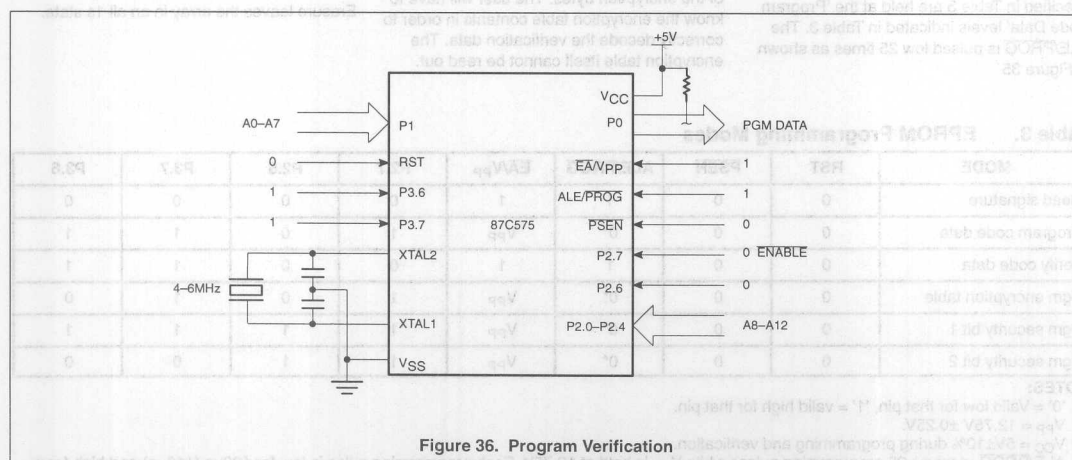
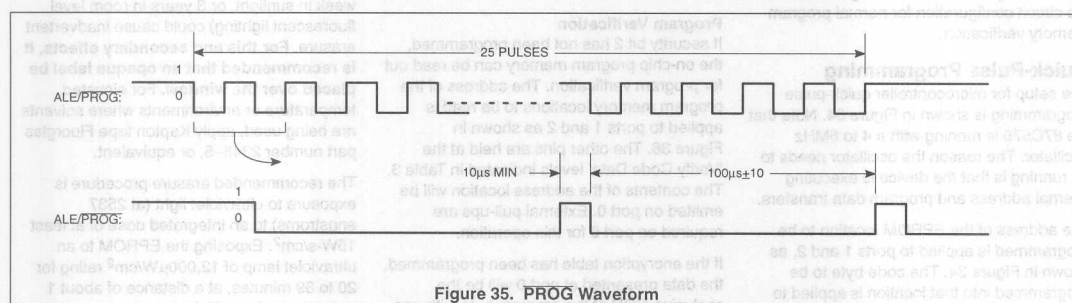
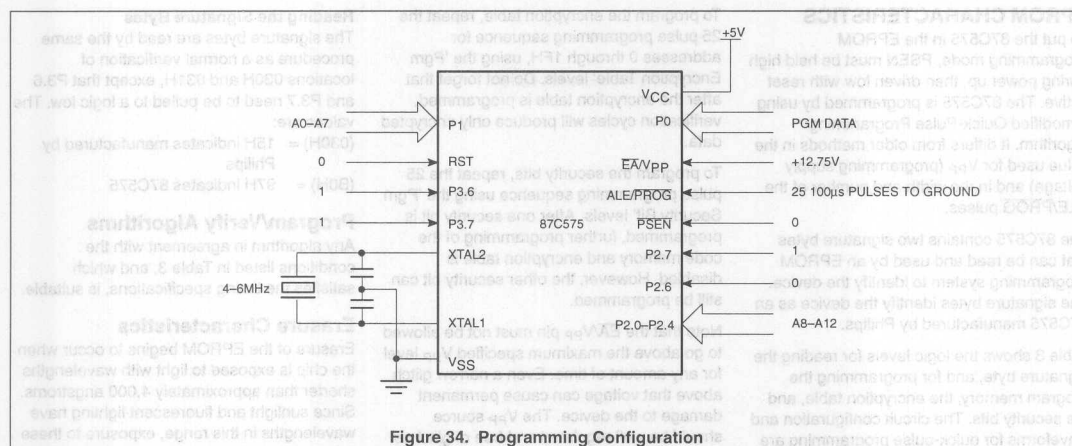
3. V<sub>CC</sub> = 5V±10% during programming and verification.

\* ALE/PROG receives 25 programming pulses while V<sub>PP</sub> is held at 12.75V. Each programming pulse is low for 100µs (±10µs) and high for a minimum of 10µs.

™Trademark phrase of Intel Corporation.

## CMOS single-chip 8-bit microcontroller

## 80C575/83C575/87C575

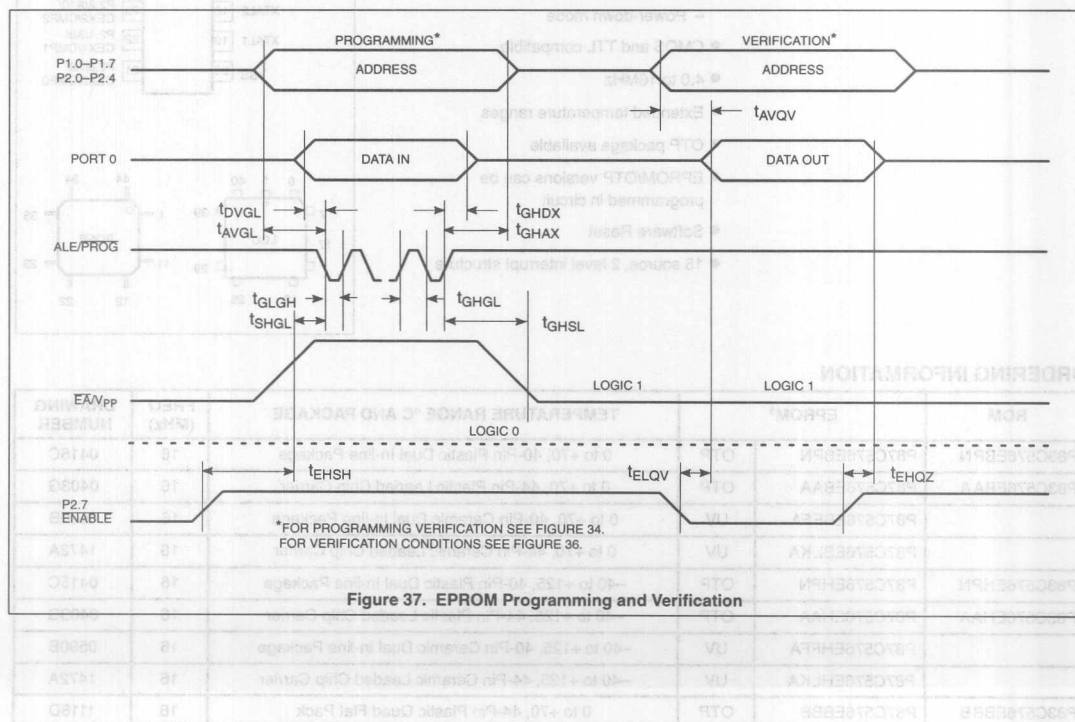


## CMOS single-chip 8-bit microcontroller 80C575/83C575/87C575

## EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

T<sub>amb</sub> = 21°C to +27°C, V<sub>CC</sub> = 5V±10%, V<sub>SS</sub> = 0V (See Figure 37)

SYMBOL	PARAMETER	MIN	MAX	UNIT
V <sub>PP</sub>	Programming supply voltage	12.5	13.0	V
I <sub>PP</sub>	Programming supply current		50	mA
1/t <sub>CLCL</sub>	Oscillator frequency	4	6	MHz
t <sub>AVGL</sub>	Address setup to PROG low	48t <sub>CLCL</sub>		
t <sub>GHAX</sub>	Address hold after PROG	48t <sub>CLCL</sub>		
t <sub>DVGL</sub>	Data setup to PROG low	48t <sub>CLCL</sub>		
t <sub>GHDX</sub>	Data hold after PROG	48t <sub>CLCL</sub>		
t <sub>ESH</sub>	P2.7 (ENABLE) high to V <sub>PP</sub>	48t <sub>CLCL</sub>		
t <sub>SHGL</sub>	V <sub>PP</sub> setup to PROG low	10		µs
t <sub>GHSL</sub>	V <sub>PP</sub> hold after PROG	10		µs
t <sub>GLGH</sub>	PROG width	90	110	µs
t <sub>AVQV</sub>	Address to data valid		48t <sub>CLCL</sub>	
t <sub>ELQZ</sub>	ENABLE low to data valid		48t <sub>CLCL</sub>	
t <sub>EHQZ</sub>	Data float after ENABLE	0	48t <sub>CLCL</sub>	
t <sub>GHGL</sub>	PROG high to PROG low	10		µs



# CMOS single-chip 8-bit microcontroller 83C576/87C576

## DESCRIPTION

The Philips 83C576/87C576 is a high-performance microcontroller fabricated with Philips high-density CMOS technology. The Philips CMOS technology combines the high speed and density characteristics of HMOS with the low power attributes of CMOS. Philips epitaxial substrate minimizes latch-up sensitivity.

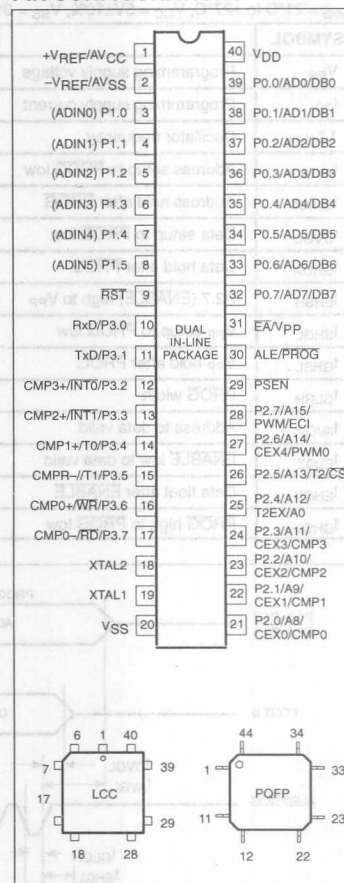
The 83C576 contains an 8k × 8 ROM (83C576) EPROM (87C576), a 256 × 8 RAM, 32 I/O lines, three 16-bit counter/timers, a Programmable Counter Array (PCA), a 10-bit, 6 channel A/D, 2 PWM outputs, an 8-bit UPI interface, a fifteen-source, two-priority level nested interrupt structure, an enhanced UART, four analog comparators, power-fail detect and oscillator fail detect circuits, and on-chip oscillator and clock circuits.

In addition, the 83C576 has a low active reset, and the port pins are reset to a low level. There is also a fully configurable watchdog timer, and internal power on clear circuit. The part includes idle mode and power-down mode states for reduced power consumption.

## FEATURES

- 80C51 based architecture
  - 8k × 8 ROM (83C576)
  - 8k × 8 EPROM (87C576)
  - 256 × 8 RAM
  - 10-bit, 6 channel A/D
  - Three 16-bit counter/timers
  - 2 PWM outputs
  - Programmable Counter Array
  - Universal Peripheral Interface
  - Enhanced UART
  - Oscillator fail detect
  - Low active reset
  - Asynchronous low port reset
  - Schmitt trigger inputs
  - 4 analog comparators
  - Watchdog timer
  - Low V<sub>CC</sub> detect
- Memory addressing capability
  - 64k ROM and 64k RAM
- Power control modes:
  - Idle mode
  - Power-down mode
- CMOS and TTL compatible
- 4.0 to 16MHz
- Extended temperature ranges
- OTP package available
- EPROM/OTP versions can be programmed in circuit
- Software Reset
- 15 source, 2 level interrupt structure

## PIN CONFIGURATIONS



## ORDERING INFORMATION

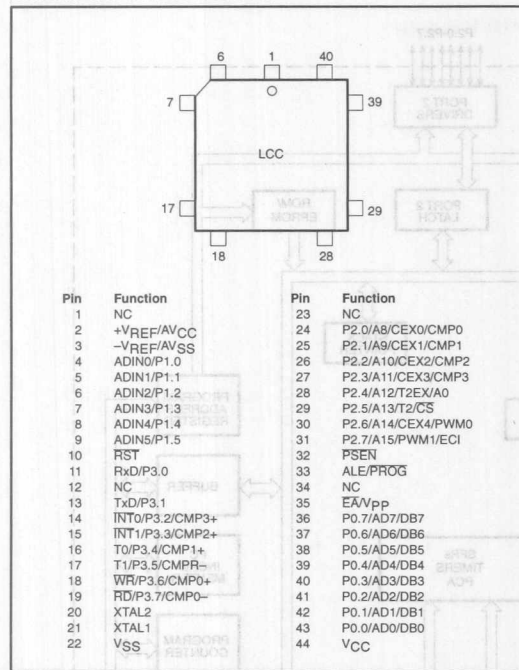
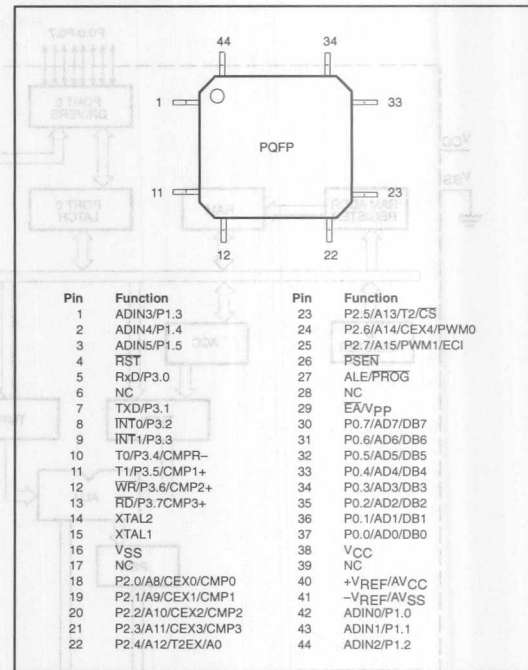
ROM	EPROM <sup>1</sup>		TEMPERATURE RANGE °C AND PACKAGE	FREQ (MHz)	DRAWING NUMBER
P83C576EBPN	P87C576EBPN	OTP	0 to +70, 40-Pin Plastic Dual In-line Package	16	0415C
P83C576EBAA	P87C576EBAA	OTP	0 to +70, 44-Pin Plastic Leaded Chip Carrier	16	0403G
	P87C576EBFFA	UV	0 to +70, 40-Pin Ceramic Dual In-line Package	16	0590B
	P87C576EBLKA	UV	0 to +70, 44-Pin Ceramic Leaded Chip Carrier	16	1472A
P83C576EHPN	P87C576EHPN	OTP	-40 to +125, 40-Pin Plastic Dual In-line Package	16	0415C
P83C576EHAA	P87C576EHAA	OTP	-40 to +125, 44-Pin Plastic Leaded Chip Carrier	16	0403G
	P87C576EHFFA	UV	-40 to +125, 40-Pin Ceramic Dual In-line Package	16	0590B
	P87C576EHLKA	UV	-40 to +125, 44-Pin Ceramic Leaded Chip Carrier	16	1472A
P83C576EBBB	P87C576EBBB	OTP	0 to +70, 44-Pin Plastic Quad Flat Pack	16	1118D

### NOTE:

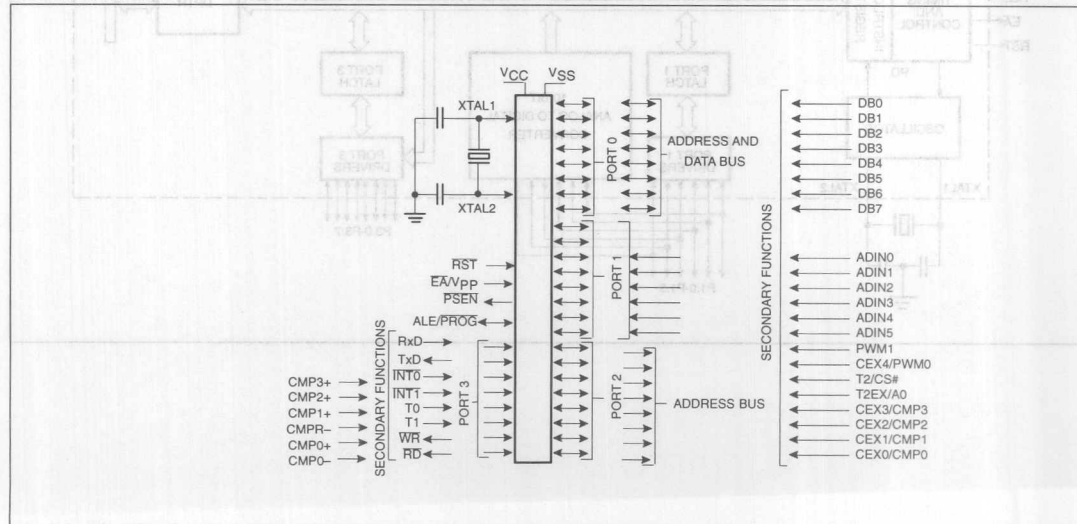
1. OTP - One Time Programmable EPROM. UV - Erasable EPROM



## CMOS single-chip 8-bit microcontroller 83C576/87C576

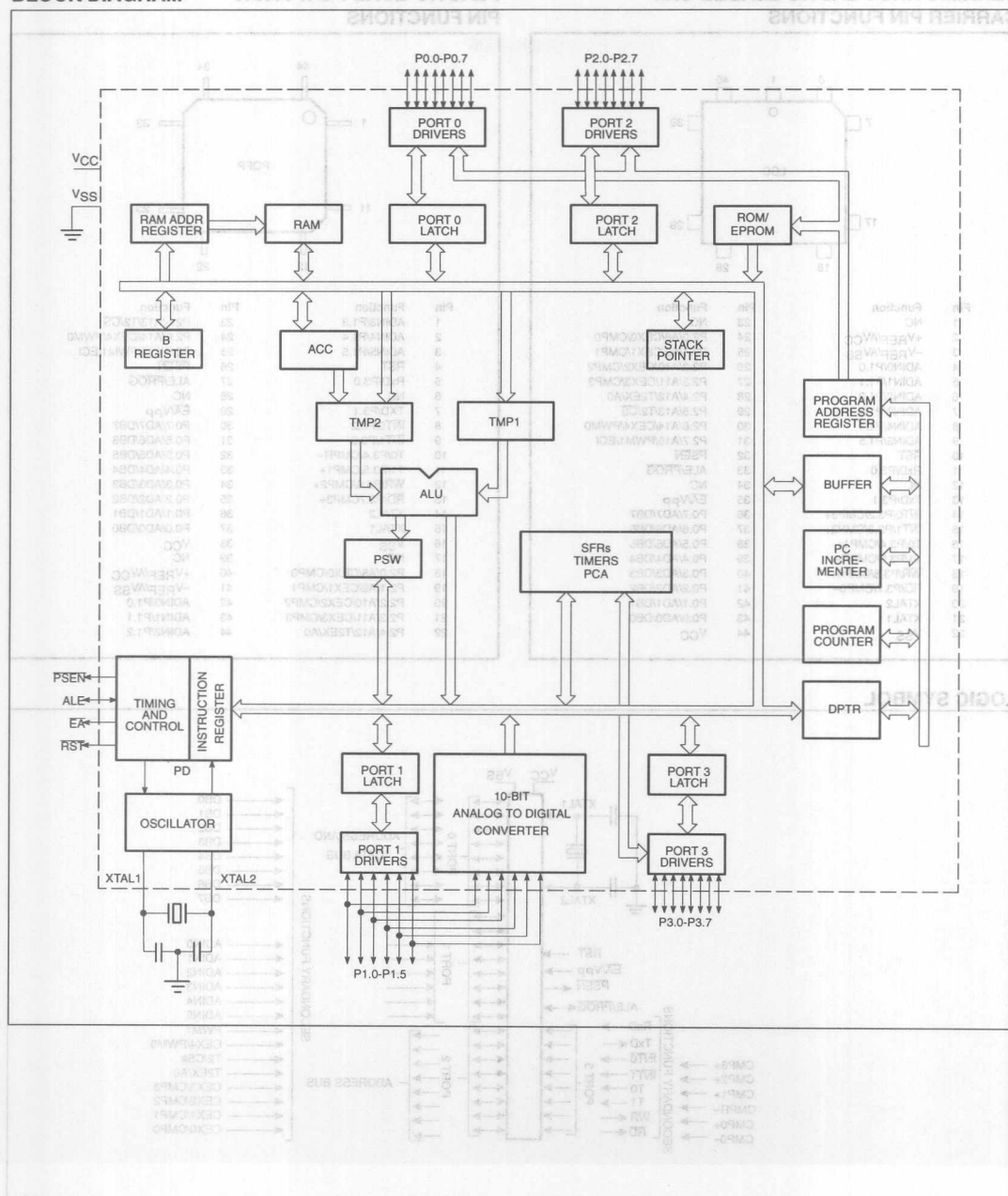
CERAMIC AND PLASTIC LEADED CHIP  
CARRIER PIN FUNCTIONSPLASTIC QUAD FLAT PACK  
PIN FUNCTIONS

## LOGIC SYMBOL



## CMOS single-chip 8-bit microcontroller 83C576/87C576

## BLOCK DIAGRAM



## CMOS single-chip 8-bit microcontroller 83C576/87C576

## PIN DESCRIPTIONS

MNEMONIC	PIN NUMBER			TYPE	NAME AND FUNCTION																																										
	DIP	LCC	QFP																																												
V <sub>SS</sub>	20	22	16	I	Ground: 0V reference.																																										
V <sub>CC</sub>	40	44	38	I	<b>Power Supply:</b> This is the power supply voltage for normal, idle, and power-down operation.																																										
P0.0-0.7	39-32	43-36	37-30	I/O	<b>Port 0:</b> Port 0 is an open-drain bidirectional I/O port. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s. Port 0 also receives code bytes during EPROM programming and outputs code bytes during program verification. External pull-ups are required during program verification. During reset, the port 0 pulldowns are turned on asynchronously, and the port register is loaded with 0's. Port 0 has 4 output modes selected on a per bit basis by writing to the P0MOD and P0OE Special Function Registers as follows: <table><tr><th>P0MOD.x</th><th>P0OE.x</th><th>Mode Description</th></tr><tr><td>0</td><td>0</td><td>Open drain (default). See Note 1.</td></tr><tr><td>0</td><td>1</td><td>Weak pullup. See Note 2.</td></tr><tr><td>1</td><td>0</td><td>High impedance. See Note 3.</td></tr><tr><td>1</td><td>1</td><td>Push-pull. See Note 4.</td></tr></table> Port 0 is also the data I/O port for the Universal Peripheral Interface (UPI). When the UPI is enabled, the port 0 drivers are normally disabled. Input/Output through P0 is controlled by pin CS, WR, RD, and A. Output is push-pull when enabled.	P0MOD.x	P0OE.x	Mode Description	0	0	Open drain (default). See Note 1.	0	1	Weak pullup. See Note 2.	1	0	High impedance. See Note 3.	1	1	Push-pull. See Note 4.																											
P0MOD.x	P0OE.x	Mode Description																																													
0	0	Open drain (default). See Note 1.																																													
0	1	Weak pullup. See Note 2.																																													
1	0	High impedance. See Note 3.																																													
1	1	Push-pull. See Note 4.																																													
P1.0-P1.5	3-8	5-9	42-44 1-3	I/O	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port. Port 1 receives the low-order address byte during program memory verification and EPROM programming. During reset, port 1 is configured as a high impedance analog input port. Digital push-pull outputs are enabled by writing 1's to the P1OE (port 1 output enable) register. The programmer must take care to prevent digital outputs from switching while an A/D conversion is in progress. Port 1 pins also serve alternate functions as follows: <table><tr><th></th><th>+V<sub>REF</sub>/AV<sub>CC</sub></th><th>-V<sub>REF</sub>/AV<sub>SS</sub></th></tr><tr><td>1</td><td>2</td><td>40</td></tr><tr><td>2</td><td>3</td><td>41</td></tr><tr><td>3</td><td>4</td><td>42</td></tr><tr><td>4</td><td>5</td><td>43</td></tr><tr><td>5</td><td>6</td><td>44</td></tr><tr><td>6</td><td>7</td><td>1</td></tr><tr><td>7</td><td>8</td><td>2</td></tr><tr><td>8</td><td>9</td><td>3</td></tr></table>		+V <sub>REF</sub> /AV <sub>CC</sub>	-V <sub>REF</sub> /AV <sub>SS</sub>	1	2	40	2	3	41	3	4	42	4	5	43	5	6	44	6	7	1	7	8	2	8	9	3															
	+V <sub>REF</sub> /AV <sub>CC</sub>	-V <sub>REF</sub> /AV <sub>SS</sub>																																													
1	2	40																																													
2	3	41																																													
3	4	42																																													
4	5	43																																													
5	6	44																																													
6	7	1																																													
7	8	2																																													
8	9	3																																													
P2.0-P2.7	21-28	24-31	18-25	I/O	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 emits the high-order address byte during accesses to external program and data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. Port 2 receives the high-order address byte during program verification and EPROM programming. During reset, the port 2 pulldowns are turned on synchronously, and the port register is loaded with 0's. Port 2 has the following output modes which can be selected on a per bit basis by writing to P2MA and P2MB: <table><tr><th>P2M1</th><th>P2M2</th><th>Mode Description</th></tr><tr><td>0</td><td>0</td><td>Open drain (default). See Note 1.</td></tr><tr><td>0</td><td>1</td><td>Weak pullup. See Note 2.</td></tr><tr><td>1</td><td>0</td><td>High impedance. See Note 3.</td></tr><tr><td>1</td><td>1</td><td>Push-pull. See Note 4.</td></tr></table> Port 2 pins serve alternate functions as follows: <table><tr><th></th><th>CEX0</th><th>CMP0</th></tr><tr><td>21</td><td>24</td><td>18</td></tr><tr><td>22</td><td>25</td><td>19</td></tr><tr><td>23</td><td>26</td><td>20</td></tr><tr><td>24</td><td>27</td><td>21</td></tr><tr><td>25</td><td>28</td><td>22</td></tr><tr><td>26</td><td>29</td><td>23</td></tr><tr><td>27</td><td>30</td><td>24</td></tr><tr><td>28</td><td>31</td><td>25</td></tr></table>	P2M1	P2M2	Mode Description	0	0	Open drain (default). See Note 1.	0	1	Weak pullup. See Note 2.	1	0	High impedance. See Note 3.	1	1	Push-pull. See Note 4.		CEX0	CMP0	21	24	18	22	25	19	23	26	20	24	27	21	25	28	22	26	29	23	27	30	24	28	31	25
P2M1	P2M2	Mode Description																																													
0	0	Open drain (default). See Note 1.																																													
0	1	Weak pullup. See Note 2.																																													
1	0	High impedance. See Note 3.																																													
1	1	Push-pull. See Note 4.																																													
	CEX0	CMP0																																													
21	24	18																																													
22	25	19																																													
23	26	20																																													
24	27	21																																													
25	28	22																																													
26	29	23																																													
27	30	24																																													
28	31	25																																													

## CMOS single-chip 8-bit microcontroller 83C576/87C576

## PIN DESCRIPTIONS (Continued)

MNEMONIC	PIN NUMBER			TYPE	NAME AND FUNCTION																																																			
	DIP	LCC	QFP																																																					
P3.0-P3.7	10-17	11, 13-19	5, 7-13	I/O	<p><b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins except P3.1 that have 1s written to them can be used as inputs but will source current when externally pulled low (see DC Electrical Characteristics: <math>I_{IL}</math>). P3.1 will be a high impedance pin except while transmitting serial data, in which case the strong pull-up will remain on continuously when outputting a 1 level. The P3.1 output drive level when transmitting can be set to one of two levels by the writing to the P3.1 register bit. During reset all pins (except P3.1) will be asynchronously driven low and will remain low until written to by software. All port 3 pins have Schmitt trigger inputs with 200mV hysteresis, except P3.2 and P3.3, which have 50mV hysteresis. Port 3 has the following output modes which can be selected on a per bit basis by writing to P3M1 and P3M2:</p> <table><tr><th>P3M1</th><th>P3M2</th><th>Mode Description</th></tr><tr><td>0</td><td>0</td><td>Open drain. See Note 1.</td></tr><tr><td>0</td><td>1</td><td>Weak pullup (default). See Note 2.</td></tr><tr><td>1</td><td>0</td><td>High impedance. See Note 3.</td></tr><tr><td>1</td><td>1</td><td>Push-pull. See Note 4.</td></tr></table> <p>Port 3 pins serve alternate functions as follows:</p> <table><tr><td>P3.0</td><td>RxD</td><td>Serial receive port</td></tr><tr><td>P3.1</td><td>TxD</td><td>Serial transmit port (enabled only when transmitting serial data)</td></tr><tr><td>P3.2</td><td>INT0</td><td>External interrupt 0</td></tr><tr><td>P3.3</td><td>INT1</td><td>External interrupt 1</td></tr><tr><td>P3.4</td><td>CMP3+</td><td>Comparator 3 positive input</td></tr><tr><td>P3.4</td><td>T0</td><td>Timer/counter 0 input</td></tr><tr><td>P3.4</td><td>CMP1+</td><td>Comparator 1 positive input</td></tr><tr><td>P3.5</td><td>T1</td><td>Timer/counter 1 input</td></tr><tr><td>P3.6</td><td>WR</td><td>External data memory write strobe</td></tr><tr><td>P3.6</td><td>CMP0+</td><td>Comparator 0 positive input</td></tr><tr><td>P3.7</td><td>RD</td><td>External data memory read strobe</td></tr><tr><td>P3.7</td><td>CMP0-</td><td>Comparator 0 negative input</td></tr></table>	P3M1	P3M2	Mode Description	0	0	Open drain. See Note 1.	0	1	Weak pullup (default). See Note 2.	1	0	High impedance. See Note 3.	1	1	Push-pull. See Note 4.	P3.0	RxD	Serial receive port	P3.1	TxD	Serial transmit port (enabled only when transmitting serial data)	P3.2	INT0	External interrupt 0	P3.3	INT1	External interrupt 1	P3.4	CMP3+	Comparator 3 positive input	P3.4	T0	Timer/counter 0 input	P3.4	CMP1+	Comparator 1 positive input	P3.5	T1	Timer/counter 1 input	P3.6	WR	External data memory write strobe	P3.6	CMP0+	Comparator 0 positive input	P3.7	RD	External data memory read strobe	P3.7	CMP0-	Comparator 0 negative input
P3M1	P3M2	Mode Description																																																						
0	0	Open drain. See Note 1.																																																						
0	1	Weak pullup (default). See Note 2.																																																						
1	0	High impedance. See Note 3.																																																						
1	1	Push-pull. See Note 4.																																																						
P3.0	RxD	Serial receive port																																																						
P3.1	TxD	Serial transmit port (enabled only when transmitting serial data)																																																						
P3.2	INT0	External interrupt 0																																																						
P3.3	INT1	External interrupt 1																																																						
P3.4	CMP3+	Comparator 3 positive input																																																						
P3.4	T0	Timer/counter 0 input																																																						
P3.4	CMP1+	Comparator 1 positive input																																																						
P3.5	T1	Timer/counter 1 input																																																						
P3.6	WR	External data memory write strobe																																																						
P3.6	CMP0+	Comparator 0 positive input																																																						
P3.7	RD	External data memory read strobe																																																						
P3.7	CMP0-	Comparator 0 negative input																																																						
RST	9	10	4	I	<p><b>Reset:</b> A low on this pin asynchronously resets all port pins to a low state except P3.1. The pin must be held low with the oscillator running for 24 oscillator cycles to initialize the internal registers. An internal diffused resistor to <math>V_{CC}</math> permits a power on reset using only an external capacitor to <math>V_{SS}</math>. RST has a Schmitt trigger input stage to provide additional noise immunity with a slow rising input voltage.</p>																																																			
ALE/PROG	30	33	27	I/O	<p><b>Address Latch Enable/Program Pulse:</b> Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory. ALE is switched off if the bit 0 in the AUXR register (8EH) is set. This pin is also the program pulse input (PROG) during EPROM programming.</p>																																																			
PSEN	29	32	26	O	<p><b>Program Store Enable:</b> The read strobe to external program memory. When the device is executing code from the external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory. PSEN is not activated during fetches from internal program memory.</p>																																																			
EAV <sub>PP</sub>	31	35	29	I	<p><b>External Access Enable/Programming Supply Voltage:</b> EAV must be externally held low to enable the device to fetch code from external program memory locations 0000H to 1FFFH. If EAV is held high, the device executes from internal program memory unless the program counter contains an address greater than 1FFFH. This pin also receives the 12.75V programming supply voltage (<math>V_{PP}</math>) during EPROM programming.</p>																																																			
XTAL1	19	21	15	I	<p><b>Crystal 1:</b> Input to the inverting oscillator amplifier and input to the internal clock generator circuits.</p>																																																			
XTAL2	18	20	14	O	<p><b>Crystal 2:</b> Output from the inverting oscillator amplifier.</p>																																																			

## NOTES:

- When Open Drain mode is selected, ports 0 and 2 have weak pulldowns to guarantee positive leakage current (see DC electrical characteristic  $I_{IL}$ ).
- When Weak Pullup mode is selected, ports bits that have 1's written to them can be used as inputs but will source current when externally pulled low (see DC electrical characteristic  $I_{IL}$ ).
- When High Impedance mode is selected, all pullups and pulldowns are turned off. The only current sourced or sunk by the pin is the parasitic leakage current (see DC electrical characteristic  $I_{L2}$  or  $I_{LC}$ , as applicable).
- When Push-Pull mode is selected, strong pullups are on continuously when emitting 1's (see DC electrical characteristic  $V_{OH}$ ).

## CMOS single-chip 8-bit microcontroller 80C576/87C576

Table 1. 87C576 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB						LSB		
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
ADC0H	A/D Channel 0 MSB	AAH									00H
ADC1H	A/D Channel 1 MSB	ABH									00H
ADC2H	A/D Channel 2 MSB	ACH									00H
ADC3H	A/D Channel 3 MSB	ADH									00H
ADC4H	A/D Channel 4 MSB	AEH									00H
ADC5H	A/D Channel 5 MSB	AFH									00H
ADC0L	A/D Channel 0 2-LSBits	9AH									00H
ADC1L	A/D Channel 1 2-LSBits	9BH									00H
ADC2L	A/D Channel 2 2-LSBits	9CH									00H
ADC3L	A/D Channel 3 2-LSBits	9CH									00H
ADC4L	A/D Channel 4 2-LSBits	9EH									00H
ADC5L	A/D Channel 5 2-LSBits	9FH									00H
ADCON	A/D Control	B1H	ADF	ADCE	AD8M	AMOD1	AMOD0	ASCA2	ASCA1	ASCA0	00H
ADCS	A/D Channel Select	B2H									00H
AUXR#	Auxiliary	8EH					RST		LO	AO	xxxxxx00B
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
CCAP0H#	Module 0 Capture High	FAH									xxxxxxxxB
CCAP1H#	Module 1 Capture High	FBH									xxxxxxxxB
CCAP2H#	Module 2 Capture High	FCH									xxxxxxxxB
CCAP3H#	Module 3 Capture High	FDH									xxxxxxxxB
CCAP4H#	Module 4 Capture High	FEH									xxxxxxxxB
CCAP0L#	Module 0 Capture Low	EAH									xxxxxxxxB
CCAP1L#	Module 1 Capture Low	EBH									xxxxxxxxB
CCAP2L#	Module 2 Capture Low	ECH									xxxxxxxxB
CCAP3L#	Module 3 Capture Low	EDH									xxxxxxxxB
CCAP4L#	Module 4 Capture Low	EEH									xxxxxxxxB
CCAPM0#	Module 0 Mode	DAH	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM1#	Module 1 Mode	DBH	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM2#	Module 2 Mode	DCH	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM3#	Module 3 Mode	DDH	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM4#	Module 4 Mode	DEH	—	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCON*#	PCA Counter Control	D8H	DF	DE	DD	DC	DB	DA	D9	D8	
CH#	PCA Counter High	F9H	CF	CR	—	CCF4	CCF3	CCF2	CCF1	CCF0	00x00000B
CL#	PCA Counter Low	E9H									00H
CMOD#	PCA Counter Mode	D9H	CIDL	WDTE	—	—	—	CPS1	CPS0	ECF	00xxx000B
			EF	EE	ED	EC	EB	EA	E9	E8	
CMP*#	Comparator	E8H	EC3DP	EC2DP	EC1DP	EC0DP	C3RO	C2RO	C1RO	C0RO	00H
CMPE#	Comparator Enable	91H	EC3TDC	EC2TDC	EC1TDC	EC0TDC	EC3OD	EC2OD	EC1OD	EC0OD	00H
DPTR:	Data Pointer (2 bytes)										
DPH	Data Pointer High	83H									00H
DPL	Data Pointer Low	82H									00H

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.



## CMOS single-chip 8-bit microcontroller 83C576/87C576

Table 1. 87C576 Special Function Registers (Continued)

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB				LSB				
IE*	Interrupt Enable	A8H	AF	AE	AD	AC	AB	AA	A9	A8	00H
			EA	EC	ET2	ES	ET1	EX1	ET0	EX0	
IP*	Interrupt Priority	B8H	BF	BE	BD	BC	BB	BA	B9	B8	x0000000B
			—	PPC	PT2	PS	PT1	PX1	PT0	PX0	
P0*	Port 0	80H	87	86	85	84	83	82	81	80	00H
			AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	
P1*	Port 1	90H	97	96	95	94	93	92	91	90	00H
			CEX4	CEX3	CEX2	CEX1	CEX0	EX1	T2EX	T2	
P2*	Port 2	A0H	A7	A6	A5	A4	A3	A2	A1	A0	00H
			AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8	
P3*	Port 3	B0H	B7	B6	B5	B4	B3	B2	B1	B0	00H
			RD	WR	T1	T0	INT1	INT0	TxD	RxD	
P0M1	Port 0 Output Mode 1	84H									00H
P0M2	Port 0 Output Mode 2	85H									00H
P1M1	Port 1 Output Mode 1	94H									FFH
P1M2	Port 1 Output Mode 2	95H									00H
P2M1	Port 2 Output Mode 1	A4H									00H
P2M2	Port 2 Output Mode 2	A5H									FFH
P3M1	Port 3 Output Mode 1	B4H									00H
P3M2	Port 3 Output Mode 2	B5H									FFH
PCON	Power Control	87H	SMOD1	SMOD0	OSF1	POF1	LVF1	WDT0F	PD	IDL	00xxx000B
PSW*	Program Status Word	D0H	D7	D6	D5	D4	D3	D2	D1	D0	00H
			CY	AC	F0	RS1	RS0	OV	—	P	
PWCON	PWM Control	BCH									00H
PWMP	PWM Prescaler	BDH									00H
PWM0	PWM Register 0	BEH									00H
PWM1	PWM Register 1	BFH									00H
RACAP2H#	Timer 2 Capture High	CBH									00H
RACAP2L#	Timer 2 Capture Low	CAH									00H
SADDR#	Slave Address	A9H									00H
SADEN#	Slave Address Mask	B9H									00H
SBUF	Serial Data Buffer	99H	9F	9E	9D	9C	9B	9A	99	98	xxxxxxxB
			SM0	SM1	SM2	REN	TB8	RB8	TI	RI	
SCON*	Serial Control	98H	8F	8E	8D	8C	8B	8A	89	88	00H
SP	Stack Pointer	81H									07H
TCON*	Timer Control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IE0	00H

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.

1. Reset value depends on reset source.

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.

1. Reset value depends on reset source.

## CMOS single-chip 8-bit microcontroller 87C576/87C576

Table 1. 87C576 Special Function Registers (Continued)

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB							LSB	
T2CON*	Timer 2 Control	C8H	CF	CE	CD	CC	CB	CA	C9	C8	00H
T2MOD#	Timer 2 Mode Control	C9H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	00000000B
TH0	Timer High 0	8CH									00H
TH1	Timer High 1	8DH									00H
TH2#	Timer High 2	CDH									00H
TL0	Timer Low 0	8AH									00H
TL1	Timer Low 1	8BH									00H
TL2#	Timer Low 2	CCH									00H
TMOD	Timer Mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H
WDCON*#	Watchdog Timer Control	C0H	C7	C6	C5	C4	C3	C2	C1	C0	11111101B
WDL#	Watchdog Timer Reload	C1H	PRE2	PRE1	PRE0	LVRE	OFRE	WDRUN	WDT0F	WDMOD	00H
WFEED1#	Watchdog Feed 1	C2H									xxH
WFEED2#	Watchdog Feed 2	C3H									xxH

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.

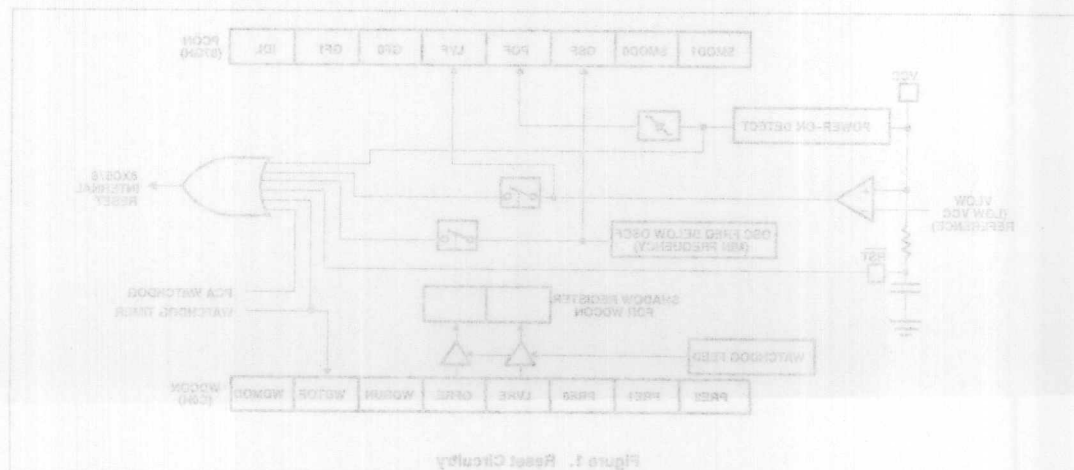
1. Reset value depends on reset source.

WDON SFR) then executing a watchdog  
 sequence (ASH to WFEED). In addition  
 there is a flag (LVF) that is set if a low voltage  
 condition is detected. The LVF flag is set  
 even if the low voltage detection circuitry is  
 disabled. Hence the low voltage  
 detection circuitry does not drive the RST#  
 pin so the LVF flag is the only way that the  
 microcontroller can determine if it has been  
 reset due to a low voltage condition.

An on-chip Oscillator Fail Detect circuit sets  
 the Oscillator Fail Flag (PCON.5) if the  
 oscillator frequency drops below OSCF for  
 one or more cycles (see A/D Electrical  
 Characteristics: OSCF) and resets the  
 87C576 if the Oscillator Fail Reset Enable bit  
 (WDON.3) is set. If OFRE is cleared, the  
 reset is disabled but OSCF will be set if the  
 oscillator fails. The RST pin is not driven by  
 the oscillator fail detect circuit. The OSCF can  
 be read by software to determine that an  
 oscillator failure has occurred. The OSCF can  
 be set or cleared by software.

## OSCILLATOR FAIL DETECT

oscillator failure has occurred. The OSCF can  
 be read by software to determine that an  
 oscillator failure has occurred. The OSCF can  
 be set or cleared by software.



## CMOS single-chip 8-bit microcontroller 83C576/87C576

**POWER ON CLEAR/  
POWER ON FLAG**

An on-chip Power On Detect Circuit resets the 8XC576 and sets the Power Off Flag (PCON.4) on power up or if  $V_{CC}$  drops to zero momentarily. The POF can only be cleared by software. The RST pin is not driven by the power on detect circuit. The POF can be read by software to determine that a power failure has occurred and can also be set by software.

**LOW VOLTAGE DETECT**

An on-chip Low Voltage Detect circuit sets the Low Voltage Flag (PCON.3) if  $V_{CC}$  drops below  $V_{LOW}$  (see DC Electrical Characteristics) and resets the 8XC576 if the Low Voltage Reset Enable bit (WDCON.4) is set. If the LVRE is cleared, the reset is disabled but LVF will still be set if  $V_{CC}$  is low. The RST pin is not driven by the low voltage detect circuit. The LVF can be read by software to determine that  $V_{CC}$  was low. The LVF can be set or cleared by software.

**OSCILLATOR FAIL DETECT**

An on-chip Oscillator Fail Detect circuit sets the Oscillator Fail Flag (PCON.5) if the oscillator frequency drops below OSCF for one or more cycles (see AC Electrical Characteristics: OSCF) and resets the 8XC576 if the Oscillator Fail Reset Enable bit (WDCON.3) is set. If OFRE is cleared, the reset is disabled but OSF will still be set if the oscillator fails. The RST pin is not driven by the oscillator fail detect circuit. The OSF can be read by software to determine that an oscillator failure has occurred. The OSF can be set or cleared by software.

**LOW ACTIVE RESET**

One of the most notable features on this part is the low active reset. At this time this is the only 80C51 derivative available that has low active reset. This feature makes it easier to interface the 8XC576 into an application to accommodate the power-on and low voltage conditions that can occur. The low active reset operates exactly the same as high active reset with the exception that the part is put into the reset mode by applying a low level to the reset pin. For power-on reset it is also necessary to invert the power-on reset circuit; connecting the 8.2K resistor from the reset pin to  $V_{CC}$  and the 10 $\mu$ f capacitor from the reset pin to ground. Figure 1 shows all of the reset related circuitry.

When reset the port pins on the 87C576 are driven low asynchronously. This is different from all other 80C51 derivatives.

The 8XC576 also has Low voltage detection circuitry that will, if enabled, force the part to reset when  $V_{CC}$  (on the part) fails below a set level. Low Voltage Reset is enabled by a normal reset. Low Voltage Reset can be disabled by clearing LVRE (bit 4 in the WDCON SFR) then executing a watchdog feed sequence (ASH to WFEED1 followed immediately by 5A to WFEED2). In addition there is a flag (LVF) that is set if a low voltage condition is detected. The LVF flag is set even if the Low Voltage detection circuitry is disabled. Notice that the Low voltage detection circuitry does not drive the RST# pin so the LVF flag is the only way that the microcontroller can determine if it has been reset due to a low voltage condition.

The 8XC576 has an on-chip power-on detection circuit that sets the POF (PCON.4) flag on power up or if the  $V_{CC}$  level momentarily drops to 0V. This flag can be used to determine if the part is being started from a power-on (cold start) or if a reset has occurred due to another condition (warm start).

**TIMERS**

The 87C576 has four on-chip timers.

Timers 0 and 1 are identical in every way to Timers 0 and 1 on the 80C51.

Timer 2 on the 8XC576 is identical to the 80C52 Timer 2 (described in detail in the 80C52 overview) with the exception that it is an up or down counter. To configure the Timer to count down the DCEN bit in the T2MOD special function register must be set and a low level must be present on the T2EX pin (P1.1).

The Watchdog timer operation and implementation is the same as that for the 8XC550 (described in the 8XC550 overview) with the exception that the reset values of the WDCON and WDL special function registers have been changed. The changes in these registers cause the watchdog timer to be enabled with a timeout of  $98304 \times T_{OSC}$  when the part is reset. The watchdog can be disabled by executing a valid feed sequence and then clearing WDRUN (bit 2 in the WDCON SFR).

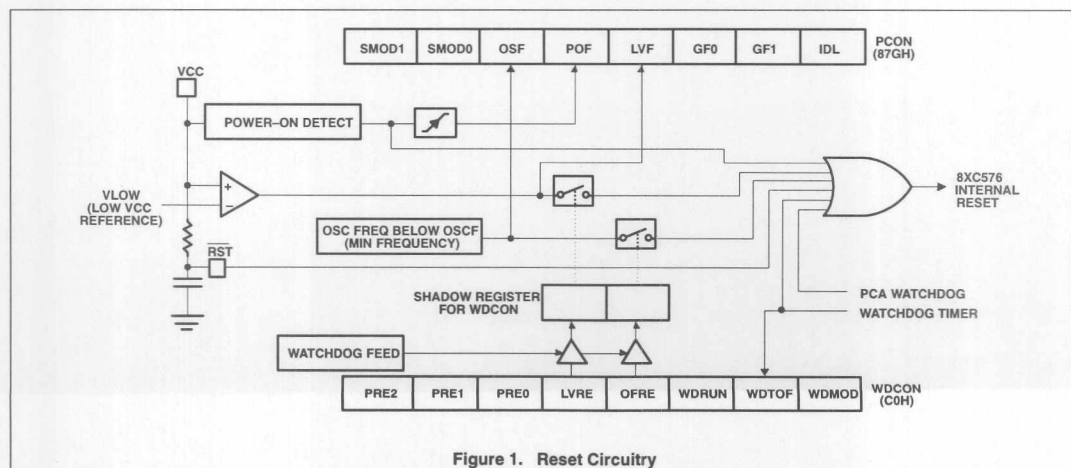


Figure 1. Reset Circuitry

## CMOS single-chip 8-bit microcontroller 83C576/87C576

**PROGRAMMABLE COUNTER ARRAY (PCA)**

The Programmable Counter Array is a special Timer that has five 16-bit capture/compare modules associated with it. Each of the modules can be programmed to operate in one of four modes: rising and/or falling edge capture, software timer, high-speed output, or pulse width modulator. Each module has a pin associated with it in port 1. Module 0 is connected to P1.3(CEX0), module 1 to P1.4(CEX1), etc. The basic PCA configuration is shown in Figure 2.

The PCA timer is a common time base for all five modules and can be programmed to run at: 1/12 the oscillator frequency, 1/4 the oscillator frequency, the Timer 0 overflow, or the input on the ECI pin (P1.2). The timer count source is determined from the CPS1 and CPS0 bits in the CMOD SFR as follows (see Figure 3):

CPS1	CPS0	PCA Timer Count Source
0	0	1/12 oscillator frequency
0	1	1/4 oscillator frequency
1	0	Timer 0 overflow
1	1	External Input at ECI pin

In the CMOD SFR are three additional bits associated with the PCA. They are CIDL which allows the PCA to stop during idle mode, WDTE which enables or disables the watchdog function on module 4, and ECF which when set causes an interrupt and the PCA overflow flag CF (in the CCON SFR) to be set when the PCA timer overflows. These functions are shown in Figure 3.

The watchdog timer function is implemented in module 4 as implemented in other parts that have a PCA that are available on the market. However, if a watchdog timer is required in the target application, it is recommended to use the hardware watchdog timer that is implemented on the 87C576 separately from the PCA (see Figure 14).

The CCON SFR contains the run control bit for the PCA and the flags for the PCA timer (CF) and each module (refer to Figure 6). To run the PCA the CR bit (CCON.6) must be set by software. The PCA is shut off by clearing this bit. The CF bit (CCON.7) is set when the PCA counter overflows and an interrupt will be generated if the ECF bit in the CMOD register is set. The CF bit can only be cleared by software. Bits 0 through 4 of the CCON register are the flags for the modules (bit 0 for module 0, bit 1 for module 1, etc.) and are set by hardware when either a match or a capture occurs. These flags also can only be cleared by software. The PCA interrupt system shown in Figure 4.

Each module in the PCA has a special function register associated with it. These registers are: CCAPM0 for module 0, CCAPM1 for module 1, etc. (see Figure 7). The registers contain the bits that control the mode that each module will operate in. The EECF bit (CCAPMn.0 where n=0, 1, 2, 3, or 4 depending on the module) enables the CCF flag in the CCON SFR to generate an interrupt when a match or compare occurs in the associated module. PWM (CCAPMn.1) enables the pulse width modulation mode. The TOG bit (CCAPMn.2) when set causes the CEX output associated with the module to toggle when there is a match between the PCA counter and the module's capture/compare register. The match bit MAT (CCAPMn.3) when set will cause the CCFn bit in the CCON register to be set when there is a match between the PCA counter and the module's capture/compare register.

The next two bits CAPN (CCAPMn.4) and CAPP (CCAPMn.5) determine the edge that a capture input will be active on. The CAPN bit enables the negative edge, and the CAPP bit enables the positive edge. If both bits are set both edges will be enabled and a capture will occur for either transition. The last bit in the register ECOM (CCAPMn.6) when set

enables the comparator function. Figure 8 shows the CCAPMn settings for the various PCA functions.

There are two additional registers associated with each of the PCA modules. They are CCAPnH and CCAPnL and these are the registers that store the 16-bit count when a capture occurs or a compare should occur. When a module is used in the PWM mode these registers are used to control the duty cycle of the output.

**PCA Capture Mode**

To use one of the PCA modules in the capture mode either one or both of the CCAPM bits CAPN and CAPP for that module must be set. The external CEX input for the module (on port 1) is sampled for a transition. When a valid transition occurs the PCA hardware loads the value of the PCA counter registers (CH and CL) into the module's capture registers (CCAPnL and CCAPnH). If the CCFn bit for the module in the CCON SFR and the ECCFn bit in the CCAPMn SFR are set then an interrupt will be generated. Refer to Figure 9.

**16-bit Software Timer Mode**

The PCA modules can be used as software timers by setting both the ECOM and MAT bits in the modules CCAPMn register. The PCA timer will be compared to the module's capture registers and when a match occurs an interrupt will occur if the CCFn (CCON SFR) and the ECCFn (CCAPMn SFR) bits for the module are both set (see Figure 10).

**High Speed Output Mode**

In this mode the CEX output (on port 1) associated with the PCA module will toggle each time a match occurs between the PCA counter and the module's capture registers. To activate this mode the TOG, MAT, and ECOM bits in the module's CCAPMn SFR must be set (see Figure 11).

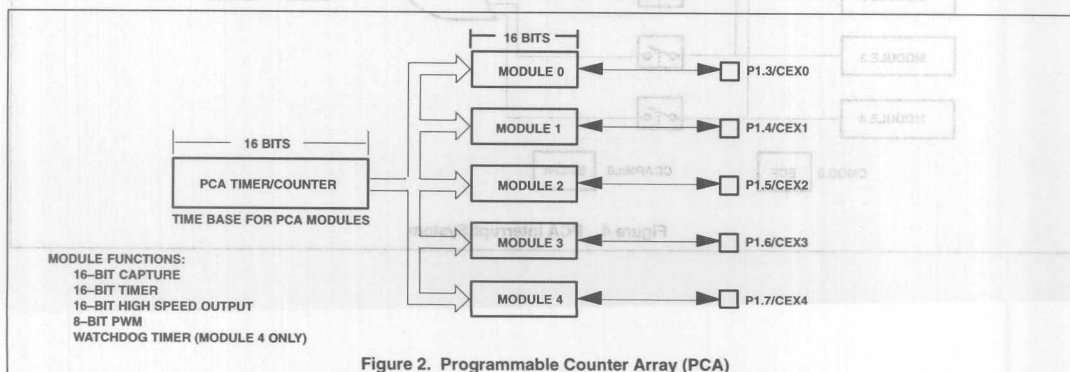
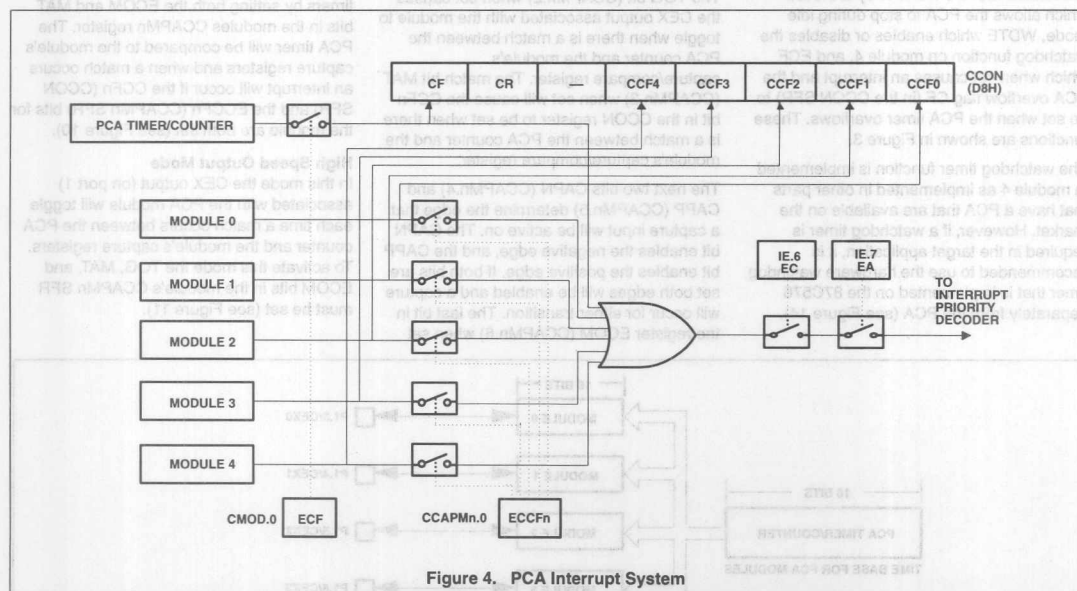
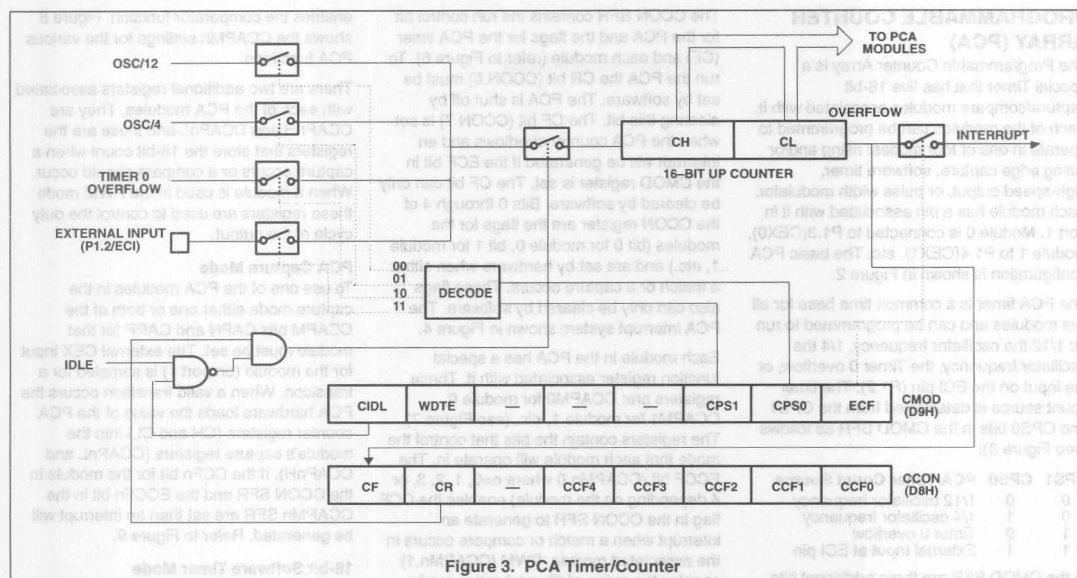


Figure 2. Programmable Counter Array (PCA)

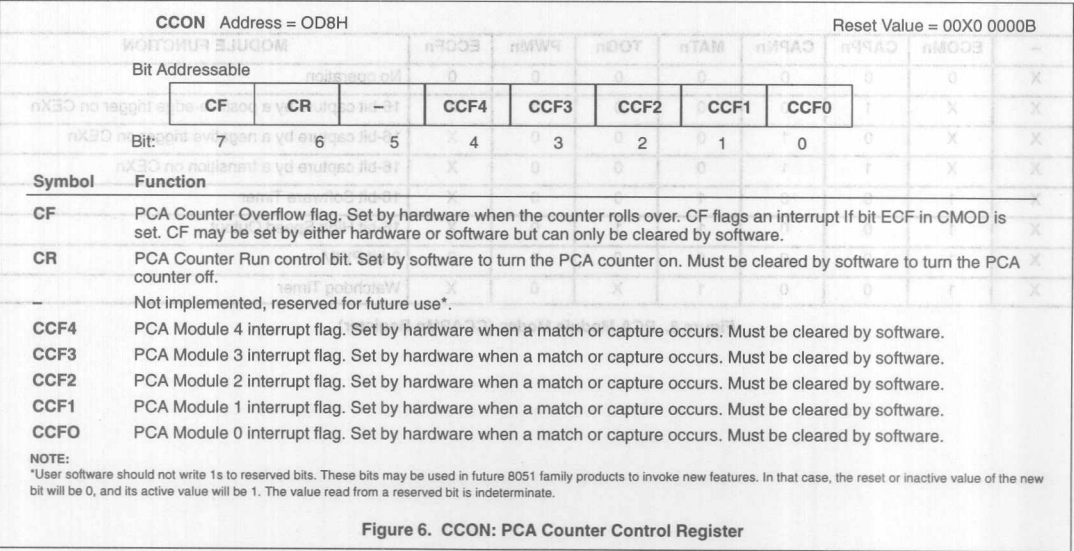
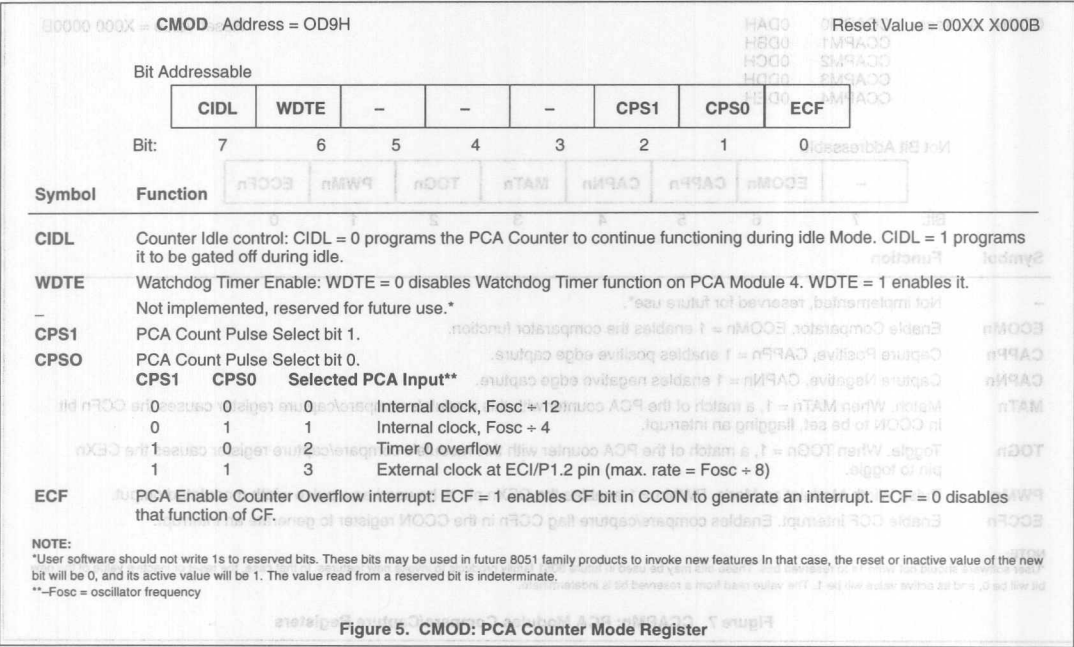
## CMOS single-chip 8-bit microcontroller

83C576/87C576

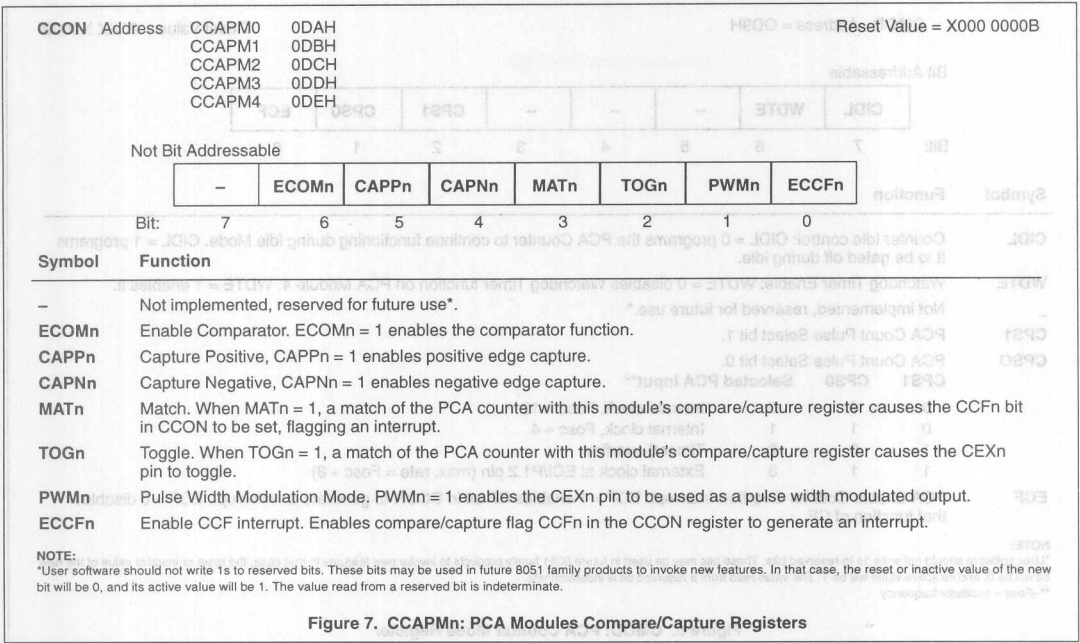




CMOS single-chip 8-bit microcontroller 83C576/87C576



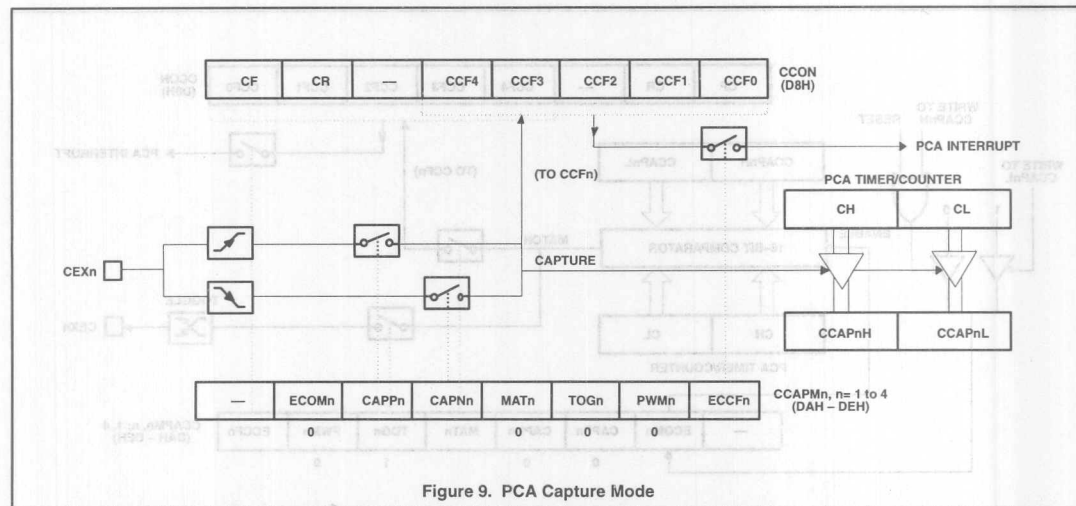
CMOS single-chip 8-bit microcontroller 83C576/87C576



–	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn	MODULE FUNCTION
X	0	0	0	0	0	0	0	No operation
X	X	1	0	0	0	0	X	16-bit capture by a positive-edge trigger on CEXn
X	X	0	1	0	0	0	X	16-bit capture by a negative trigger on CEXn
X	X	1	1	0	0	0	X	16-bit capture by a transition on CEXn
X	1	0	0	1	0	0	X	16-bit Software Timer
X	1	0	0	1	1	0	X	16-bit High Speed Output
X	1	0	0	0	0	1	0	8-bit PWM
X	1	0	0	1	X	0	X	Watchdog Timer

Figure 8. PCA Module Modes (CCAPMn Register)

## CMOS single-chip 8-bit microcontroller 83C576/87C576



**Figure 9. PCA Capture Mode**

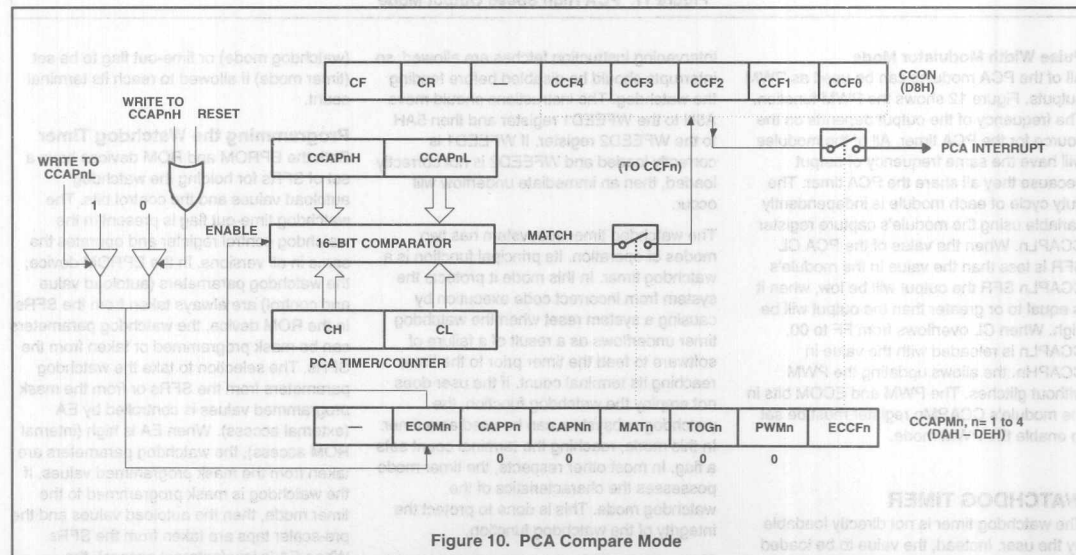


Figure 10. PCA Compare Mode

## CMOS single-chip 8-bit microcontroller 83C576/87C576

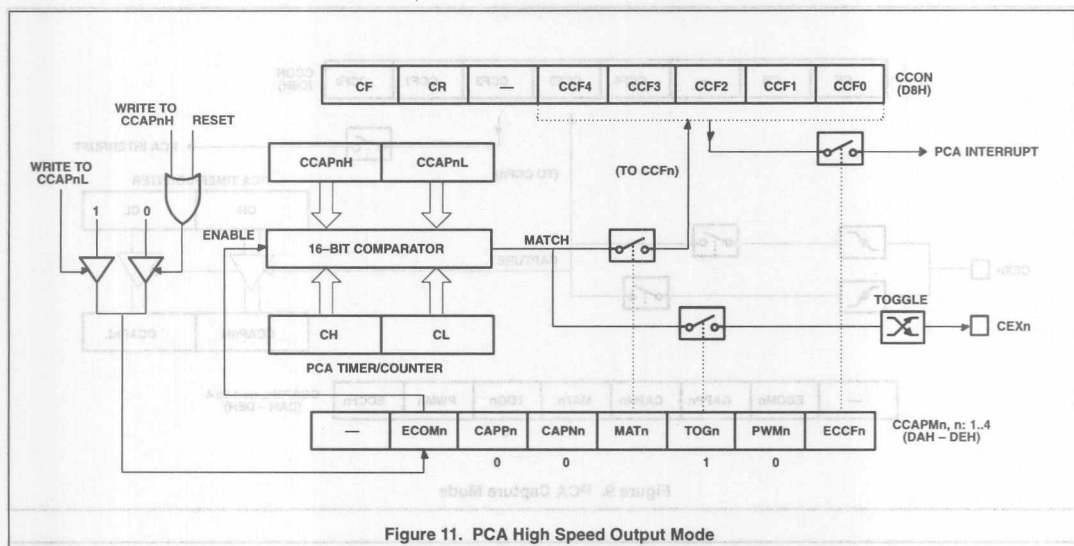


Figure 11. PCA High Speed Output Mode

### Pulse Width Modulator Mode

All of the PCA modules can be used as PWM outputs. Figure 12 shows the PWM function. The frequency of the output depends on the source for the PCA timer. All of the modules will have the same frequency of output because they all share the PCA timer. The duty cycle of each module is independently variable using the module's capture register CCAPLn. When the value of the PCA CL SFR is less than the value in the module's CCAPLn SFR the output will be low, when it is equal to or greater than the output will be high. When CL overflows from FF to 00, CCAPLn is reloaded with the value in CCAPHn. The allows updating the PWM without glitches. The PWM and ECOM bits in the module's CCAPMn register must be set to enable the PWM mode.

## WATCHDOG TIMER

The watchdog timer is not directly loadable by the user. Instead, the value to be loaded into the main timer is held in an autoloader register or is part of the mask ROM programming. In order to cause the main timer to be loaded with the appropriate value, a special sequence of software action must take place. This operation is referred to as feeding the watchdog timer.

To feed the watchdog, two instructions must be sequentially executed successfully. No

intervening instruction fetches are allowed, so interrupts should be disabled before feeding the watchdog. The instructions should move A5H to the WFEED1 register and then 5AH to the WFEED2 register. If WFEED1 is correctly loaded and WFEED2 is not correctly loaded, then an immediate underflow will occur.

The watchdog timer subsystem has two modes of operation. Its principal function is a watchdog timer. In this mode it protects the system from incorrect code execution by causing a system reset when the watchdog timer underflows as a result of a failure of software to feed the timer prior to the timer reaching its terminal count. If the user does not employ the watchdog function, the watchdog subsystem can be used as a timer. In this mode, reaching the terminal count sets a flag. In most other respects, the timer mode possesses the characteristics of the watchdog mode. This is done to protect the integrity of the watchdog function.

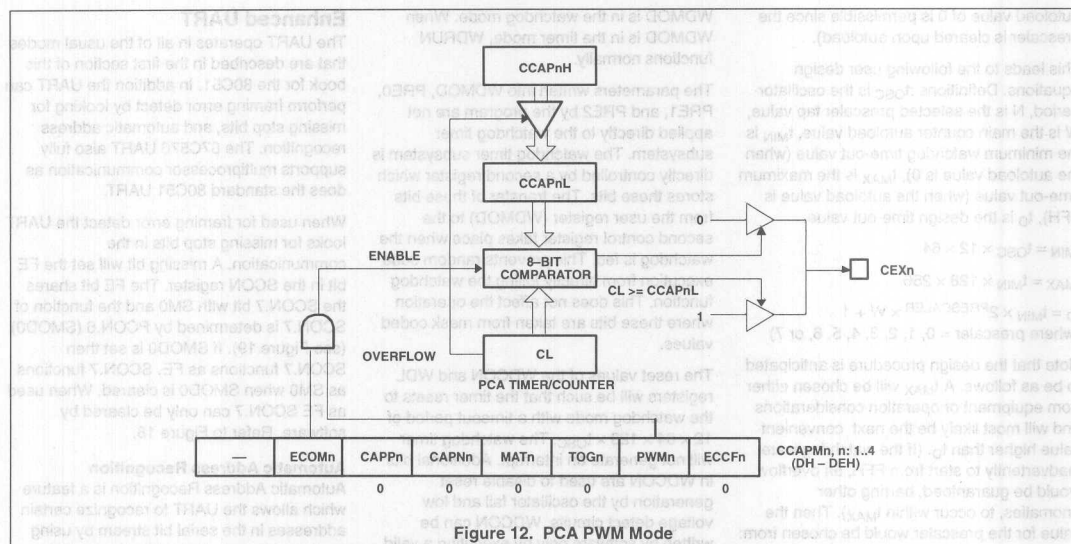
The watchdog timer subsystem consists of a prescaler and a main counter. The prescaler has 8 selectable taps off the final stages and the output of a selected tap provides the clock to the main counter. The main counter is the section that is loaded as a result of the software feeding the watchdog and it is the section that causes the system reset

(watchdog mode) or time-out flag to be set (timer mode) if allowed to reach its terminal count.

## Programming the Watchdog Timer

Both the EPROM and ROM devices have a set of SFRs for holding the watchdog autoloading values and the control bits. The watchdog time-out flag is present in the watchdog control register and operates the same in all versions. In the EPROM device, the watchdog parameters (autoload value and control) are always taken from the SFRs. In the ROM device, the watchdog parameters can be mask programmed or taken from the SFRs. The selection to take the watchdog parameters from the SFRs or from the mask programmed values is controlled by EA (external access). When EA is high (internal ROM access), the watchdog parameters are taken from the mask programmed values. If the watchdog is mask programmed to the timer mode, then the autoloading values and the pre-scaler taps are taken from the SFRs. When EA is low (external access), the watchdog parameters are taken from the SFRs. The user should be able to leave code in his program which initializes the watchdog SFRs even though he has migrated to the mask ROM part. This allows no code changes from EPROM prototyping to ROM coded production parts.

## CMOS single-chip 8-bit microcontroller 83C576/87C576



## Watchdog Detailed Operation

## EPROM Device (and ROMless Operation: EA = 0)

In the ROMless operation (ROM part, EA = 0) and in the EPROM device, the watchdog operates in the following manner (see Figure 14).

Whether the watchdog is in the watchdog or timer mode, when external RESET is applied, the following takes place:

- Watchdog mode bit set to watchdog mode.
- Watchdog run control bit set to ON.
- Autoload register set to 00 (min. count).
- Watchdog time-out flag cleared.
- Prescaler is cleared.
- Prescaler tap set to the highest divide.
- Autoload takes place.

The watchdog can be fed even though it is in the timer mode.

Note that the operational concept is for the watchdog mode of operation, when coming out of a hardware reset, the software should load the autoload registers, set the mode to watchdog, and then feed the watchdog (cause an autoload). The watchdog will now be starting at a known point.

If the watchdog is in the watchdog mode and running and happens to underflow at the time

the external RESET is applied, the watchdog time-out flag will be cleared.

When the watchdog is in the watchdog mode and the watchdog underflows, the following action takes place (see Figure 16):

- Autoload takes place.
- Watchdog time-out flag is set
- Mode bit unchanged.
- Watchdog run bit unchanged.
- Autoload register unchanged.
- Prescaler tap unchanged.
- All other device action same as external reset.

Note that if the watchdog underflows, the program counter will start from 00H as in the case of an external reset. The watchdog time-out flag can be examined to determine if the watchdog has caused the reset condition. The watchdog time-out flag bit can be cleared by software.

When the watchdog is in the timer mode and the timer software underflows, the following action takes place:

- Autoload takes place.
- Watchdog time-out flag is set
- Mode bit unchanged.
- Watchdog run bit unchanged.

- Autoload register unchanged.
- Prescaler tap unchanged.

## Mask ROM Device (EA = 1)

In the mask ROM device, the watchdog mode bit (WDMOD) is mask programmed and the bit in the watchdog command register is read only and reflects the mask programmed selection. If the mask programmed mode bit selects the timer mode, then the watchdog run bit (WDRUN) operates as described under EPROM Device. If the mask programmed bit selects the watchdog mode, then the watchdog run bit has no effect on the timer operation (see Figure 15).

## Watchdog Function

The watchdog consists of a programmable prescaler and the main timer. The prescaler derives its clock from the on-chip oscillator. The prescaler consists of a divide by 12 followed by a 13 stage counter with taps from stage 6 through stage 13. This is shown in Figure 17. The tap selection is programmable. The watchdog main counter is a down counter clocked (decremented) each time the programmable prescaler underflows. The watchdog generates an underflow signal (and is autoloaded) when the watchdog is at count 0 and the clock to decrement the watchdog occurs. The watchdog is 8 bits long and the autoload value can range from 0 to FFH. (The



## CMOS single-chip 8-bit microcontroller 83C576/87C576

autoload value of 0 is permissible since the prescaler is cleared upon autoload).

This leads to the following user design equations. Definitions:  $t_{OSC}$  is the oscillator period,  $N$  is the selected prescaler tap value,  $W$  is the main counter autoload value,  $t_{MIN}$  is the minimum watchdog time-out value (when the autoload value is 0),  $t_{MAX}$  is the maximum time-out value (when the autoload value is FFH),  $t_D$  is the design time-out value.

$$t_{MIN} = t_{OSC} \times 12 \times 64$$

$$t_{MAX} = t_{MIN} \times 128 \times 256$$

$$t_D = t_{MIN} \times 2^{\text{PRESCALER}} \times W + 1$$

(where prescaler = 0, 1, 2, 3, 4, 5, 6, or 7)

Note that the design procedure is anticipated to be as follows. A  $t_{MAX}$  will be chosen either from equipment or operation considerations and will most likely be the next convenient value higher than  $t_D$ . (If the watchdog were inadvertently to start from FFH, an overflow would be guaranteed, barring other anomalies, to occur within  $t_{MAX}$ ). Then the value for the prescaler would be chosen from:

$$\text{prescaler} = \log_2 (t_{MAX} / (t_{OSC} \times 12 \times 256)) - 6$$

This then also fixes  $t_{MIN}$ . An autoload value would then be chosen from:

$$W = t_D / t_{MIN} - 1$$

The software must be written so that a feed operation takes place every  $t_D$  seconds from the last feed operation. Some tradeoffs may need to be made. It is not advisable to include feed operations in minor loops or in subroutines unless the feed operation is a specific subroutine.

#### Watchdog Control Register (WDCON) (Bit Addressable) Address C0

The following bits of this register are read only in the ROM part when EA is high: WDMOD, PRE0, PRE1, and PRE2. That is, the register will reflect the mask programmed values. In the ROM part with EA high, these bits are taken from mask coded bits and are not readable by the program. WDRUN is read only in the ROM part when EA is high and

WDMOD is in the watchdog mode. When WDMOD is in the timer mode, WDRUN functions normally.

The parameters written into WDMOD, PRE0, PRE1, and PRE2 by the program are not applied directly to the watchdog timer subsystem. The watchdog timer subsystem is directly controlled by a second register which stores these bits. The transfer of these bits from the user register (WDMOD) to the second control register takes place when the watchdog is fed. This prevents random code execution from directly foiling the watchdog function. This does not affect the operation where these bits are taken from mask coded values.

The reset values of the WDCON and WDL registers will be such that the timer resets to the watchdog mode with a timeout period of  $12 \times 64 \times 128 \times t_{OSC}$ . The watchdog timer will not generate an interrupt. Additional bits in WDCON are used to disable reset generation by the oscillator fail and low voltage detect circuits. WDCON can be written by software only by executing a valid watchdog feed sequence.

#### WDCON Register Bit Definitions

WDCON.7	PRE2	Prescaler Select 2, reset to 1
WDCON.6	PRE1	Prescaler Select 1, reset to 1
WDCON.5	PRE0	Prescaler Select 0, reset to 1
WDCON.4	LVRE	Low Voltage Reset Enable, reset to 1 (enabled)
WDCON.3	OFRE	Oscillator Fail Reset Enable, reset to 1 (enabled)
WDCON.2	WDRUN	Watchdog Run, reset to 1 (enabled)
WDCON.1	WDT0F	Watchdog Timeout Flag, reset to 0
WDCON.0	WDMOD	Watchdog Mode, reset to 1 (watchdog mode)

#### Enhanced UART

The UART operates in all of the usual modes that are described in the first section of this book for the 80C51. In addition the UART can perform framing error detect by looking for missing stop bits, and automatic address recognition. The 87C576 UART also fully supports multiprocessor communication as does the standard 80C51 UART.

When used for framing error detect the UART looks for missing stop bits in the communication. A missing bit will set the FE bit in the SCON register. The FE bit shares the SCON.7 bit with SM0 and the function of SCON.7 is determined by PCON.6 (SMOD0) (see Figure 19). If SMOD0 is set then SCON.7 functions as FE. SCON.7 functions as SM0 when SMOD0 is cleared. When used as FE SCON.7 can only be cleared by software. Refer to Figure 18.

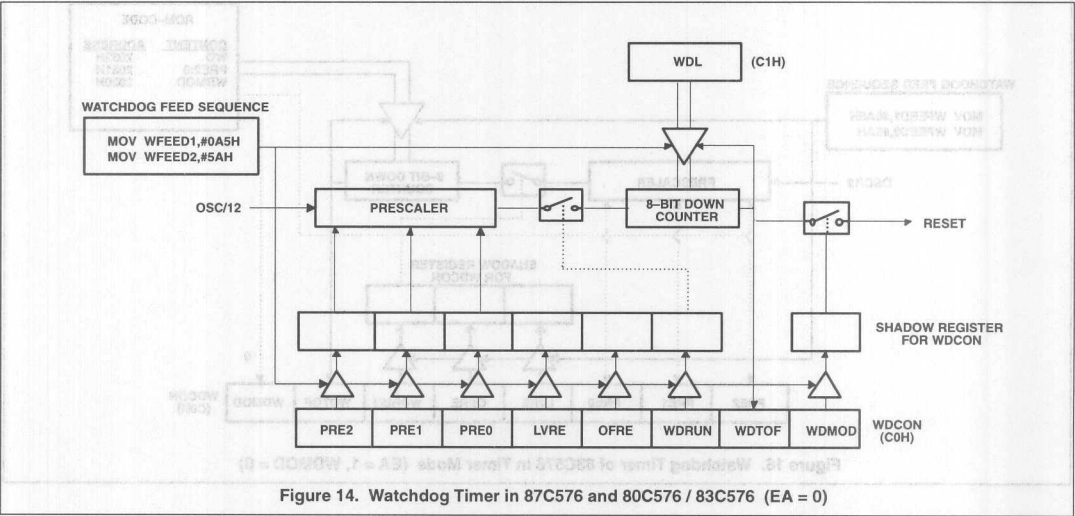
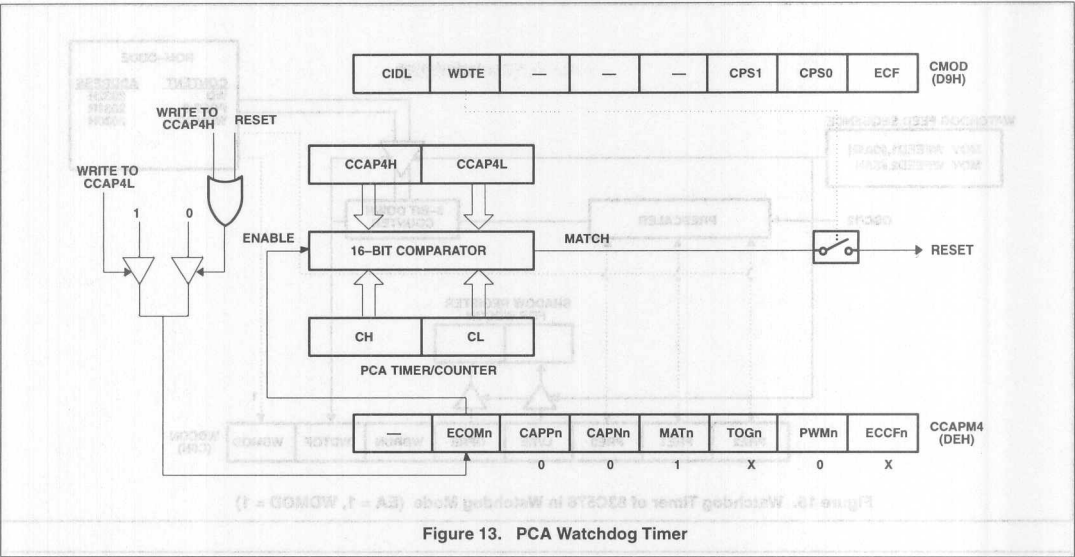
#### Automatic Address Recognition

Automatic Address Recognition is a feature which allows the UART to recognize certain addresses in the serial bit stream by using hardware to make the comparisons. This feature saves a great deal of software overhead by eliminating the need for the software to examine every serial address which passes by the serial port. This feature is enabled by setting the SM2 bit in SCON. In the 9 bit UART modes, mode 2 and mode 3, the Receive Interrupt flag (RI) will be automatically set when the received byte contains either the "Given" address or the "Broadcast" address. The 9 bit mode requires that the 9th information bit is a 1 to indicate that the received information is an address and not data. Automatic address recognition is shown in Figure 20.

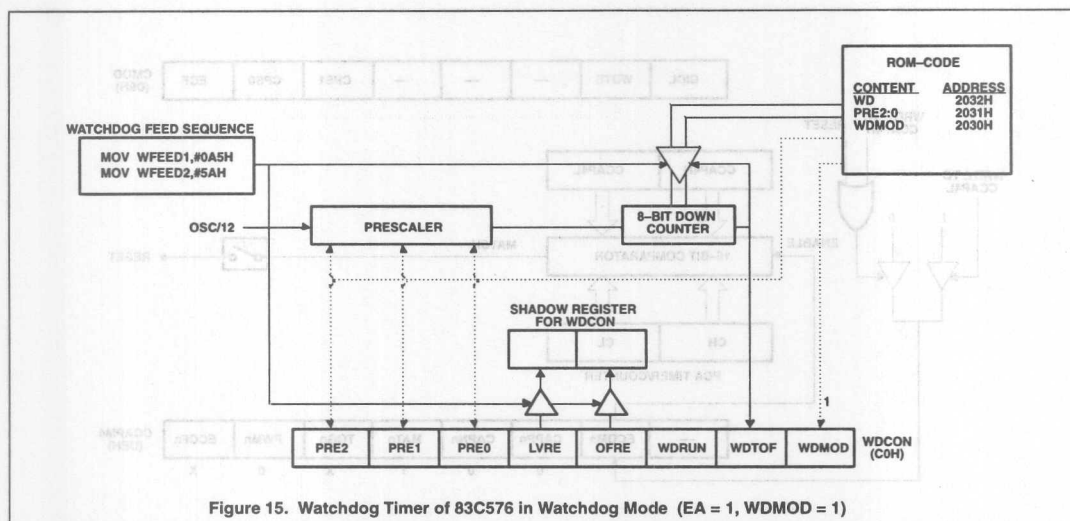
The 8 bit mode is called Mode 1. In this mode the RI flag will be set if SM2 is enabled and the information received has a valid stop bit following the 8 address bits and the information is either a Given or Broadcast address.

Mode 0 is the Shift Register mode and SM2 is ignored.

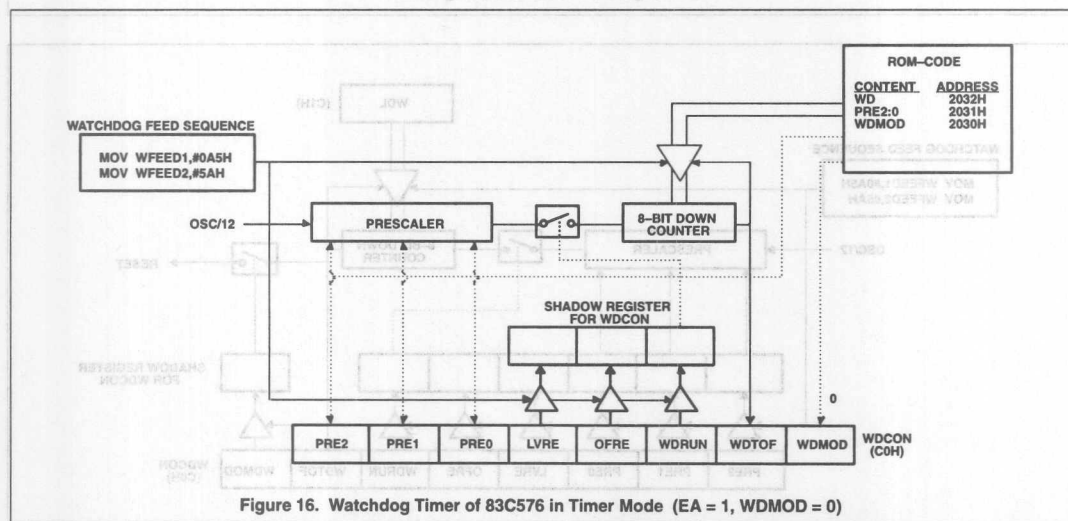
CMOS single-chip 8-bit microcontroller 83C576/87C576



## CMOS single-chip 8-bit microcontroller 83C576/87C576

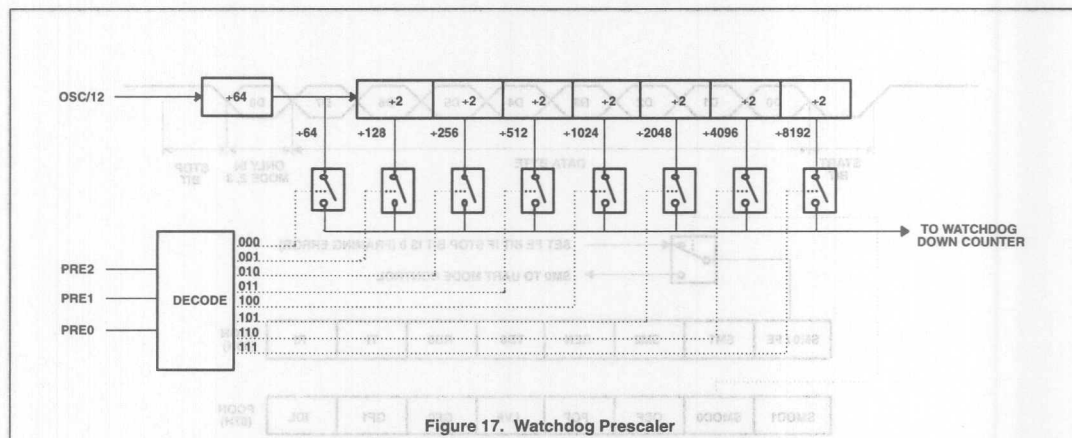


**Figure 15. Watchdog Timer of 83C576 in Watchdog Mode (EA = 1, WDMOD = 1)**



**Figure 16. Watchdog Timer of 83C576 in Timer Mode (EA = 1, WDMOD = 0)**

## CMOS single-chip 8-bit microcontroller 83C576/87C576



**Figure 17. Watchdog Prescaler**

**SCON Address = 98H** Reset Value = 0000 0000B

Bit Addressable

<b>SM0/FE</b>	<b>SM1</b>	<b>SM2</b>	<b>REN</b>	<b>TB8</b>	<b>RB8</b>	<b>TI</b>	<b>RI</b>
Bit: 7	6	5	4	3	2	1	0

(SMOD0 = 0/1)\*

Symbol	Function
<b>FE</b>	Framing Error bit. This bit is set by the receiver when an invalid stop bit is detected. The FE bit is not cleared by valid frames but should be cleared by software. The SMOD0 bit must be set to enable access to the FE bit.
<b>SM0</b>	Serial Port Mode Bit 0, (SMOD0 must = 0 to access bit SM0)
<b>SM1</b>	Serial Port Mode Bit 1
<b>SM0</b>	<b>SM1</b>
<b>Mode</b>	<b>Description</b>
0	0
0	1
1	0
1	1
<b>SM2</b>	Enables the Automatic Address Recognition feature in Modes 2 or 3. If SM2 = 1 then RI will not be set unless the received 9th data bit (RB8) is 1, indicating an address, and the received byte is a Given or Broadcast Address. In Mode 1, if SM2 = 1 then RI will not be activated unless a valid stop bit was received, and the received byte is a Given or Broadcast Address. In Mode 0, SM2 should be 0.
<b>REN</b>	Enables serial reception. Set by software to enable reception. Clear by software to disable reception.
<b>TB8</b>	The 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired.
<b>RB8</b>	In modes 2 and 3, the 9th data bit that was received. In Mode 1, if SM2 = 0, RB8 is the stop bit that was received. In Mode 0, RB8 is not used.
<b>TI</b>	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.
<b>RI</b>	Receive interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM2). Must be cleared by software.

\*SMOD0 is located at PCON6.  
\*\*Fosc = oscillator frequency

## CMOS single-chip 8-bit microcontroller 83C576/87C576

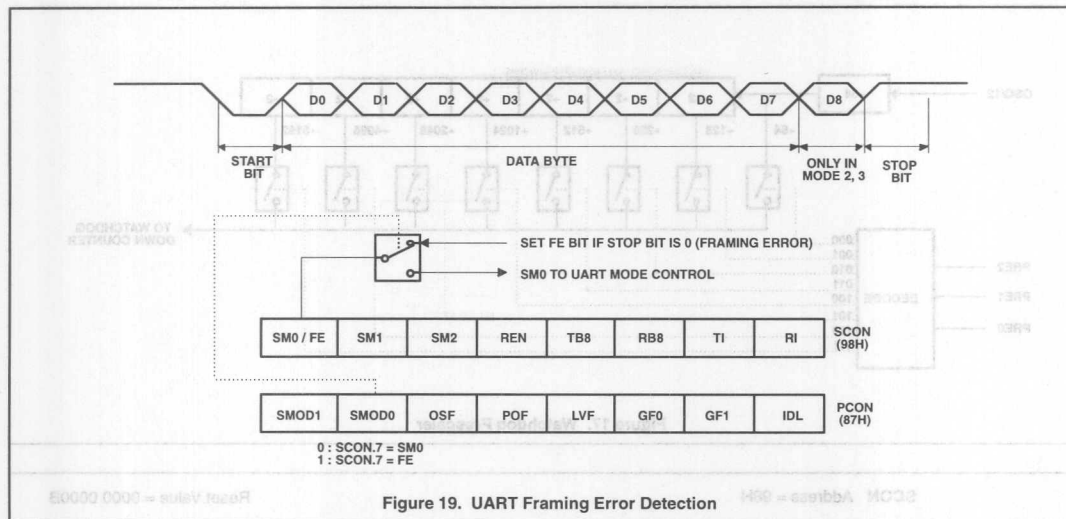


Figure 19. UART Framing Error Detection

Using the Automatic Address Recognition feature allows a master to selectively communicate with one or more slaves by invoking the Given slave address or addresses. All of the slaves may be contacted by using the Broadcast address. Two special Function Registers are used to define the slave's address, SADDR, and the address mask, SADEN. SADEN is used to define which bits in the SADDR are to be used and which bits are "don't care". The SADEN mask can be logically ANDed with the SADDR to create the "Given" address which the master will use for addressing each of the slaves. Use of the Given address allows multiple slaves to be recognized while excluding others. The following examples will help to show the versatility of this scheme:

Slave 0	SADDR = 1100 0000
	SADEN = 1111 1101
	Given = 1100 00X0
Slave 1	SADDR = 1100 0000
	SADEN = 1111 1110
	Given = 1100 000X

In the above example SADDR is the same and the SADEN data is used to differentiate between the two slaves. Slave 0 requires a 0 in bit 0 and it ignores bit 1. Slave 1 requires a 0 in bit 1 and bit 0 is ignored. A unique address for Slave 0 would be 1100 0010 since slave 1 requires a 0 in bit 1. A unique address for slave 1 would be 1100 0001 since a 1 in bit 0 will exclude slave 0. Both

slaves can be selected at the same time by an address which has bit 0 = 0 (for slave 0) and bit 1 = 0 (for slave 1). Thus, both could be addressed with 1100 0000.

In a more complex system the following could be used to select slaves 1 and 2 while excluding slave 0:

Slave 0	SADDR = 1100 0000
	SADEN = 1111 1001
	Given = 1100 0XX0
Slave 1	SADDR = 1110 0000
	SADEN = 1111 1010
	Given = 1110 0X0X
Slave 2	SADDR = 1110 0000
	SADEN = 1111 1100
	Given = 1110 00XX

In the above example the differentiation among the 3 slaves is in the lower 3 address bits. Slave 0 requires that bit 0 = 0 and it can be uniquely addressed by 1110 0110. Slave 1 requires that bit 1 = 0 and it can be uniquely addressed by 1110 and 0101. Slave 2 requires that bit 2 = 0 and its unique address is 1110 0011. To select Slaves 0 and 1 and exclude Slave 2 use address 1110 0100, since it is necessary to make bit 2 = 1 to exclude slave 2.

The Broadcast Address for each slave is created by taking the logical OR of SADDR and SADEN. Zeros in this result are treated

as don't-cares. In most cases, interpreting the don't-cares as ones, the broadcast address will be FF hexadecimal.

Upon reset SADDR (SFR address 0A9H) and SADEN (SFR address 0B9H) are loaded with 0s. This produces a given address of all "don't cares" as well as a Broadcast address of all "don't cares". This effectively disables the Automatic Addressing mode and allows the microcontroller to use standard 80C51 type UART drivers which do not make use of this feature.

### Analog Comparators

Four analog comparators are provided on chip. Three comparators have a common negative reference CMPR- and independent positive inputs CMP1+, CMP2+, CMP3+ on port 3. The fourth comparator has independent positive and negative inputs CMP0+ and CMP0- on port 1. The CMP register contains an output and enable bit for each comparator. The CMP register is bit addressable and is located at SFR address E8H. Figure 21 shows the connection of the comparators.

Pullups at the comparator input pins will be disabled by hardware when the comparator is enabled. In addition, to make inputs high impedance, the corresponding port SFR bits must be set by software to disable the pulldowns.



## CMOS single-chip 8-bit microcontroller 87C576/87C576

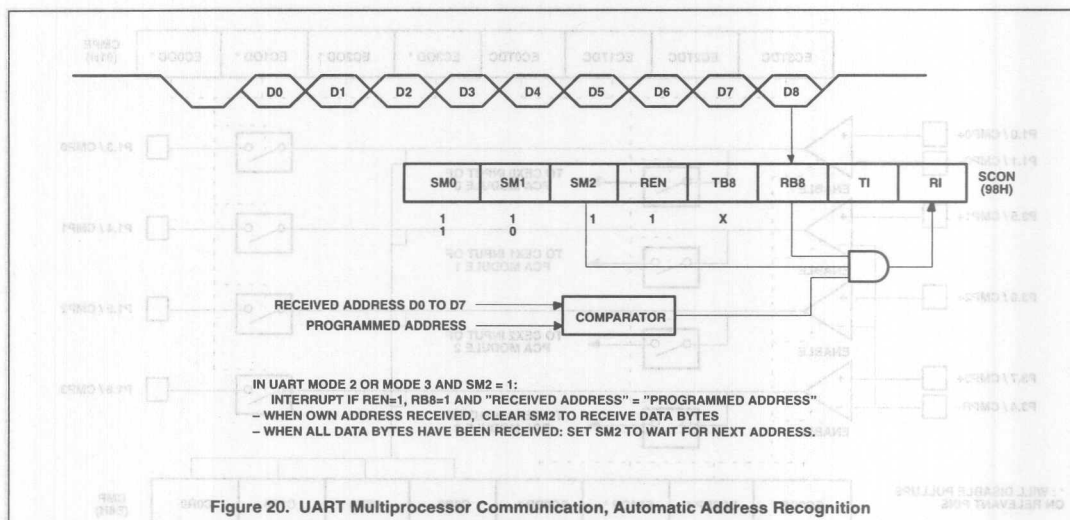


Figure 20. UART Multiprocessor Communication, Automatic Address Recognition

**CMP Register Bit Definitions**

- CMP.7 enable comparator 3, disable pullups at P3.4, P3.7
- CMP.6 enable comparator 2, disable pullups at P3.4, P3.6
- CMP.5 enable comparator 1, disable pullups at P3.4, P3.5
- CMP.4 enable comparator 0, disable pullups at P1.0, P1.1
- CMP.3 comparator 3 output (read only)
- CMP.2 comparator 2 output (read only)
- CMP.1 comparator 1 output (read only)
- CMP.0 comparator 0 output (read only)

All comparators are disabled automatically in power down mode, in idle mode unused comparators should be disabled by software to save power. A comparator can generate an interrupt that will terminate idle mode when used to drive a PCA capture input.

The CMPE register contains bits to enable each comparator to drive external output pins or internal PCA capture inputs. Pullups at the output pins are disabled by hardware when the external comparator output is enabled. The comparator output is wire-ORed with the corresponding port SFR bit, so the SFR bit must also be set by software to enable the output.

**CMPE Register Bit Definitions**

- CMPE.7 enables comparator 3 to drive CEX3
- CMPE.6 enables comparator 2 to drive CEX2
- CMPE.5 enables comparator 1 to drive CEX1
- CMPE.4 enables comparator 0 to drive CEX0
- CMPE.3 enables comparator 3 output on P1.6 (open drain)
- CMPE.2 enables comparator 2 output on P1.5 (open drain)
- CMPE.1 enables comparator 1 output on P1.4 (open drain)
- CMPE.0 enables comparator 0 output on P1.3 (open drain)

When 1s are written to CMPE bits 7-4, the comparator outputs will drive the corresponding capture input. When 1s are written to CMPE bits 3-0 the comparator output will also drive the corresponding port 1 pin. If the comparator's enabled to drive the capture input but not the port pin, then the port pin can be used for general purpose I/O. When a comparator output is enabled, pullups at the output pin are disabled and the output becomes open drain.

There are two special function registers associated with the comparators. They are CMP which contains the comparator enables and a bit that can be read by software to determine the state of each comparator's output, and CMPE which controls whether the output from each comparator drives the

associated output pin or a capture input associated with one of the PCA modules.

The CMP registers bits 0-3 can be read by software to determine the state of the output of each comparator. To do this the associated comparator must be enabled but the output in port 1 can be disabled. This allows easy polling of the comparator output value without the need to use up a port pin.

The CMPE register allows the comparator to drive the associated PCA module capture input, so that on compare a capture can be generated in the PCA. Bits 0-3 of this register enable the comparator output to drive the associated port 1 output circuitry. Used as a comparator output this circuitry is open drain. To enable the comparator output to drive to port 1, the corresponding port bit must also be set to disable the pulldown. If the comparator is not enabled to drive the port 1 circuitry, the associated port 1 pin can be used for other I/O. This includes when a comparator is enabled to drive the capture input to a PCA module.

**Reduced EMI Mode**

There are two bits in the AUXR register that can be set to reduce the internal clock drive and disable the ALE output. AO (AUXR.0) when set turns off the ALE output. LO (AUXR.1) when set reduces the drive of the internal clock circuitry. Both bits are cleared on Reset. With LO set the 87C576 will still operate at 12MHz, but will have reduced EMI in the range above 100MHz.

## CMOS single-chip 8-bit microcontroller 83C576/87C576

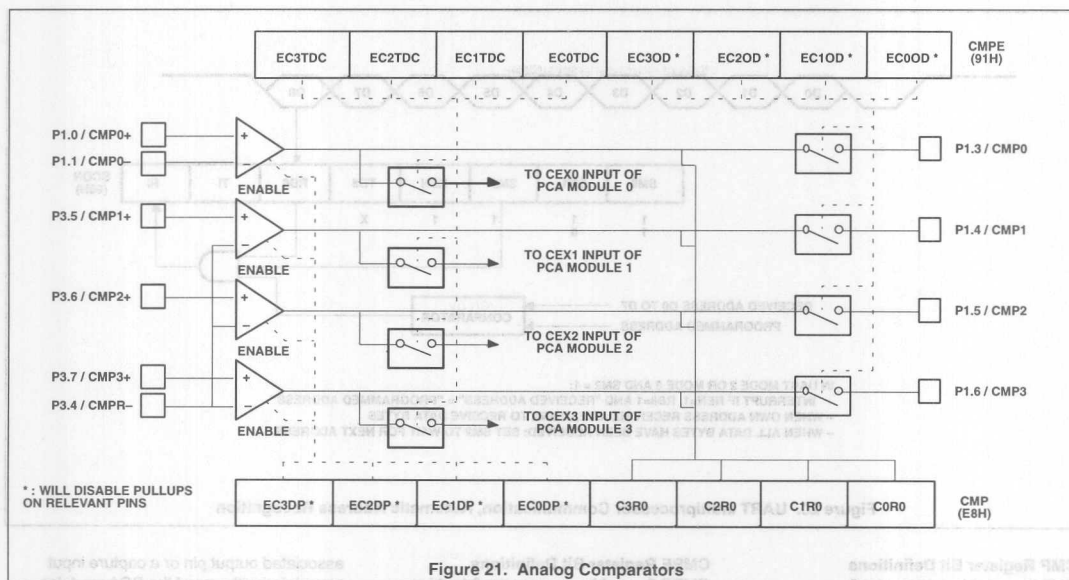


Figure 21. Analog Comparators

## 8XC576 Reduced EMI Mode

## AUXR (0X8E)

—	—	—	—	—	—	—	—	LO	AO
---	---	---	---	---	---	---	---	----	----

AO: Turns off ALE output.

LO: Reduces drive of internal clock circuitry. 8XC576 spec'd to 12MHz when LO set.

## INTERNAL RESET

Internal resets generated by the power on, low voltage, and oscillator fail detect circuits are self timed to guarantee proper initialization of the 8XC576. Reset will be held approximately 24 oscillator periods after normal conditions are detected by all enabled detect circuits. Internal resets do not drive RST but will cause missing pulses on ALE.

## Analog to Digital Converter

The 8XC576 has a 6 channel 10 bit successive approximation A/D converter with separate result registers for each channel. Operating modes are provided for single or multiple channel conversions and multiple conversions of a single channel without software intervention. The ADC can also be operated in 8 bit mode with faster conversion times. Registers ADC0H–ADC5H contain the MSBs and ADC0L–ADC5L bits 6 and 7 contain the 2 LSBs of the conversion result for each channel. The ADCS register determines which channels are converted in multiple channel modes. If the ADCS bit

corresponding to a channel is set, that channel is converted, else if the bit is clear the channel is skipped.

A/D Channel Select (ADCS) Register  
(Reset Value = 00H)

ADCS5 ADCS.5 – A/D channel 5 select bit  
ADCS4 ADCS.4 – A/D channel 4 select bit  
ADCS3 ADCS.3 – A/D channel 3 select bit  
ADCS2 ADCS.2 – A/D channel 2 select bit  
ADCS1 ADCS.1 – A/D channel 1 select bit  
ADCS0 ADCS.0 – A/D channel 0 select bit

A/D Control (ADCON) Register  
(Reset Value = 00H)

ADF ADCON.7 – A/D conversion complete flag  
ADCE ADCON.6 – A/D conversion enable  
AD8M ADCON.5 – A/D 8-bit mode  
AMOD1 ADCON.4 – A/D mode select bit 1  
AMOD0 ADCON.3 – A/D mode select bit 0  
ASCA2 ADCON.2 – A/D channel address bit 2  
ASCA1 ADCON.1 – A/D channel address bit 1  
ASCA0 ADCON.0 – A/D channel address bit 0

## AMOD1 AMOD0

0 0 Single Conversion Mode – channel selected by bits ASCA2..0 in ADCON is converted, the result placed in the associated result registers; ADF is set on completion.

0 1 Multiple Channel Scan Mode – all channels elected in the ADCS register are converted starting with the channel addressed by bits ASCA2..0 in ADCON, conversion results are placed in the corresponding result registers for each channel. ADF is set when the last conversion is completed.  
1 0 Single Channel Multiple Conversion – channel selected by bits ASCA2..0 in ADCON is converted 6 times and all 6 results are saved in ADC0H–ADC5H and ADC0L–ADC5L; ADF is set when all conversions are complete.  
1 1 Multiple Channel Continuous – same as Multiple Channel Scan mode but repeats as long as ADCE=1, ADF is set when all channels have been converted once. Hardware will prevent the ADC from writing to the result registers while they are being read.

Flag ADF is set upon completion of a conversion, if the ADC interrupt enable bit EAD is set, the program will vector to the ADC interrupt location when ADF is set.

## CMOS single-chip 8-bit microcontroller

## PWM'S

The pulse width modulator system of the 8XC576 contains two PWM output channels. These channels generate pulses of programmable length and interval. The prescaler and counter are common to both PWM channels.

The prescaler is loaded with the complement of the PWMP register during counter overflow, internal reset, and when EN/CLR# = 0. The repetition frequency is defined by the 8-bit prescaler which clocks the counter. The prescaler division factor = PWMP+1. Reading the PWMP gives the current reload value. The actual count of the prescaler cannot be read.

The 8-bit counter counts from 0–254 inclusive. The value of the counter is compared to the contents of the compare registers PWM0 and PWM1. When the counter compares to the compare register, that register's output goes LOW. When the counter reaches zero the output is set HIGH unless PWMn = 00H. The duty cycle of each channel is defined by the contents of its compare register an is in the range of 0 to 1, programmed in increments of 1/255.

The outputs can be set continuously low by loading PWMn with 00H and continuously high by loading with FFH.

The PWM counter is enabled with bit EN/CLR# of the PWCON register. Output to the port pin is separately enabled by setting the PWEn bits in the PWCON register. The counter remains active if EN/CLR# is set

even if both PWEN bits are reset. The PWM function is reset by a chip reset. In idle mode, the PWM will function as configured by PWCON. In power-down the state of the PWM will freeze when the internal clock stops. If the chip is awakened with an external interrupt, the PWM will continue to function from its state when power-down was entered. The EN/CLR# bit of PWCON will clear the counter and load the contents of the PWMP into the prescaler when set LOW. If PWEn is set at this time the output will go HIGH unless PWMn is 00H.

The repetition frequency is given by:

$$f_{PWM} = \frac{f_{OSC}}{((510 \times (1 + PWMP)))}$$

An oscillator frequency of 12MHz results in a repetition range of 92Hz to 23.5KHz.

The high/low ratio of PWMn if  $\text{PWMn}/(255-\text{PWMn})$  for PWMn values except 255. A PWMn value of 255 results in a high PWMn output.

In order for the PWMn output to be used as a standard I/O pin, PWMn must be reset. The PWM counter can still be used as an internal timer by setting EN/CLR#.

### Auxiliary Register Bit Definitions (AUXR = 8EH)

AO AUXR.0 ALE Off, when set turns off

## ALE

LO AUXR.1 Low Speed, reduces internal clock drive

AUXR.2

RST	AUXR.3	Software reset bit
-----	--------	--------------------

### Interrupt Enable 0 (IE0) Register

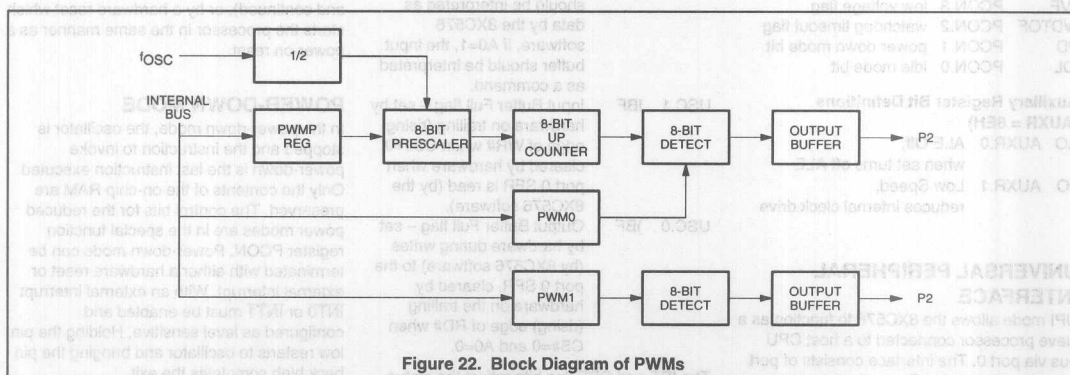
EA	IE0.7	Enable all interrupts
EC	IE0.6	Enable PCA interrupt
ET2	IE0.5	Enable Timer 2 interrupt
ES	IE0.4	Enable Serial I/O interrupt
ET1	IE0.3	Enable Timer 1 interrupt
EX1	IE0.2	Enable External interrupt 1
ET0	IE0.1	Enable Timer 0 interrupt
EX0	IE0.0	Enable External interrupt 0

### Interrupt Enable 1 (IE1) Register

EOB	IE1.7	Enable OBF interrupt
EIB	IE1.6	Enable IBF interrupt
EAD	IE1.5	Enable ADC interrupt
EC4	IE1.4	Enable PCA module 4 interrupt
EC3	IE1.3	Enable PCA module 3 interrupt
EC2	IE1.2	Enable PCA module 2 interrupt
EC1	IE1.1	Enable PCA module 1 interrupt
EC0	IE1.0	Enable PCA module 0 interrupt

### Interrupt Priority 0 (IP0) Register

	IP0.7	(reserved)
PC	IP0.6	PCA interrupt priority
PT2	IP0.5	Timer 2 interrupt priority
PS	IP0.4	Serial I/O interrupt priority
PT1	IP0.3	Timer 1 interrupt priority
PX1	IP0.2	External interrupt 1 priority
PT0	IP0.1	Timer 0 interrupt priority
PX0	IP0.0	External interrupt 0 priority



**Figure 22. Block Diagram of PWMs**

## CMOS single-chip 8-bit microcontroller 83C576/87C576

## Interrupt Priority 1 (IP1) Register

POB	IP1.7	OBF interrupt priority
PIB	IP1.6	IBF interrupt priority
PAD	IP1.5	ADC interrupt priority
PC4	IP1.4	PCA module 4 interrupt priority
PC3	IP1.3	PCA module 3 interrupt priority
PC2	IP1.2	PCA module 2 interrupt priority
PC1	IP1.1	PCA module 1 interrupt priority
PC0	IP1.0	PCA module 0 interrupt priority

Priority	Source	Flag	Vector
			highest priority
1	INT0	IE0	03H
2	ADC	ADF	3BH
3	TIMER 0	TF0	0BH
4	INT1	IE1	13H
5	TIMER 1	TF1	1BH
6	SERIAL	RI, TI	23H
7	PCA0	CC0	43H
8	PCA1	CC1	4BH
9	PCA2	CC2	53H
10	PCA3	CC3	5BH
11	PCA4	CC4	63H
12	PCA	ECF	33H
13	TIMER 2	TF2/EXF2	2BH
14	UPI	IBF	6BH
15	UPI	OBF	73H
			lowest priority

## Power Control (PCON) Register

SMOD1	PCON.7	double baud rate bit
SMOD0	PCON.6	SCON.7 access control
OSF	PCON.5	oscillator fail flag
POF	PCON.4	power off flag
LVF	PCON.3	low voltage flag
WDT0F	PCON.2	watchdog timeout flag
PD	PCON.1	power down mode bit
IDL	PCON.0	idle mode bit

## Auxiliary Register Bit Definitions (AUXR = 8EH)

AO	AUXR.0	ALE Off, when set turns off ALE
LO	AUXR.1	Low Speed, reduces internal clock drive

## UNIVERSAL PERIPHERAL INTERFACE

UPI mode allows the 8XC576 to function as a slave processor connected to a host CPU bus via port 0. The interface consists of port 0 input and output buffer registers and the UPI control/status register (UCS). UPI mode is enabled by setting the UPI enable bit (UE) in the UCS. When operating in UPI mode, port 0 pins normally float. Access to port 0 is controlled by inputs WR#, RD#, CS#, and A0. RD# and WR# are the external read and write strobes controlled by the host CPU.

CS# is the chip select input, normally a decoded address from the host CPU bus, which qualifies RD# and WR# (these pins have no effect when CS#=1). The A0 pin is an address input from the host CPU which selects either the port 0 output buffer or the UCS register to be output during a read operation. During a write operation, the value of the A0 pin is latched in the UA flag in the UCS register. The following is a summary of the UPI data control inputs:

CS#	RD#	WR#	A	
0	0	1	0	read port 0 output buffer, clear OBF
0	0	1	1	read UPI control/status register
0	1	0	0	write data to input buffer set IBF, clear AF
0	1	0	1	write command to input buffer set IBF, AF
1	x	x	x	disable input/output

## UPI Control Status Register

(UCS, Reset value = 00H)

UCS.7	ST7	User defined status bit
UCS.6	ST6	User defined status bit
UCS.5	ST5	User defined status bit
UCS.4	ST4	User defined status bit
UCS.3	UE	UPI Enable bit – if UE=1, UPI is enabled, if UE=0, UPI is disabled and port 0 functions normally.
UCS.2	AF	Address Flag – contains status of the A0 (address) pin during the last write. If A0=0, the input buffer should be interpreted as data by the 8XC576 software, if A0=1, the input buffer should be interpreted as a command.
UCS.1	IBF	Input Buffer Full flag – set by hardware on trailing (rising) edge of WR# when CS#=0, cleared by hardware when port 0 SFR is read (by the 8XC576 software).
UCS.0	IBF	Output Buffer Full flag – set by hardware during writes (by 8XC576 software) to the port 0 SFR, cleared by hardware on the trailing (rising) edge of RD# when CS#=0 and A0=0.

The IBF and OBF flag bits reflect the status of the input/output buffers. The host CPU writes to the 8XC576 by driving data on the external bus connected to port 0 and strobing the WR# pin while CS#=0. The WR# strobe latches port 0 data in the input buffer and sets the IBF flag on the trailing (rising) edge. When the 8XC576 reads from port 0 in UPI

mode, it reads from the input buffer and clears the IBF. When the 8XC576 writes to port 0 in UPI mode, it writes to the output buffer which sets the OBF. The host CPU can read the output buffer or the UCS register enabling the port 0 drivers, the OBF flag is cleared when the output buffer is read.

## OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator, as shown in the Logic Symbol, page 741.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

## IDLE MODE

In idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

## POWER-DOWN MODE

In the power-down mode, the oscillator is stopped and the instruction to invoke power-down is the last instruction executed. Only the contents of the on-chip RAM are preserved. The control bits for the reduced power modes are in the special function register PCON. Power-down mode can be terminated with either a hardware reset or external interrupt. With an external interrupt INT0 or INT1 must be enabled and configured as level sensitive. Holding the pin low restarts oscillator and bringing the pin back high completes the exit.

## DESIGN CONSIDERATIONS

At power-on, the voltage on V<sub>CC</sub> must come up with RST low for a proper start-up.

Table 2 shows the state of I/O ports during low current operating modes.



# CMOS single-chip 8-bit microcontroller 83C576/87C576

Table 2. External Pin Status During Idle and Power-Down Modes

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

## ROM CODE SUBMISSION

When submitting ROM code for the 83C576, the following must be specified:

1. 8k byte user ROM data
2. 32 byte ROM encryption key
3. ROM security bits
4. The watchdog timer parameters.

ADDRESS	CONTENT	BIT(S)	COMMENT
0000H to 1FFFH	DATA	7:0	User ROM Data
2000H to 201FH	KEY	7:0	ROM Encryption Key FFH = no encryption
2020H	SEC	0	ROM Security Bit 1
2020H	SEC	1	ROM Security Bit 2 0 = enable security 1 = disable security
2030H	WMOD	0	Watchdog mode bit; 00H = timer mode 01H = watchdog mode
2031H	PRE2:0	2:0	Watchdog prescaler selection; 00H = divide by 12 × 64 07H = divide by 12 × 64 × 128 (see specification)
2032H	WD	7:0	Watchdog autoloading value (see specification)

**Security Bit 1:** When programmed, this bit has two effects on masked ROM parts:

1. External MOV<sub>C</sub> is disabled, and
2. EA# is latched on Reset.

**Security Bit 2:** When programmed, this bit inhibits Verify User ROM.

## ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>

PARAMETER	RATING	UNIT
Operating temperature under bias	-55 to +125	°C
Storage temperature range	-65 to +150	°C
Voltage on EA/V <sub>PP</sub> pin to V <sub>SS</sub>	0 to +13.0	V
Voltage on any other pin to V <sub>SS</sub>	-0.5 to +6.5	V
Maximum I <sub>OL</sub> per I/O pin	15	mA
Power dissipation (based on package heat transfer limitations, not device power consumption)	1.5	W

### NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
2. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
3. Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V<sub>SS</sub> unless otherwise noted.



## CMOS single-chip 8-bit microcontroller 83C576/87C576

## DC ELECTRICAL CHARACTERISTICS

T<sub>amb</sub> = 0°C to +70°C and -40°C to +125°C, V<sub>CC</sub> = 5V ±20%, V<sub>SS</sub> = 0V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP <sup>1</sup>	MAX	
V <sub>IL</sub>	Input low voltage (Ports 0, 2, 3, except 3.2, 3.3)		-0.5		0.5V <sub>CC</sub> -0.6	V
V <sub>IL1</sub>	Input low voltage (Ports 1, 3.2, 3.3)		-0.5		0.65V <sub>CC</sub> -0.5	V
V <sub>IL2</sub>	Input low voltage (E <sub>A</sub> )		0		0.2V <sub>CC</sub> -0.45	V
V <sub>IL3</sub>	Input low voltage (XTAL1, RST)		-0.5		0.2V <sub>CC</sub> -0.1	V
V <sub>IH</sub>	Input high voltage (Ports 0, 2, 3, except 3.2, 3.3)		0.5V <sub>CC</sub> +0.8		V <sub>CC</sub> +0.5	V
V <sub>IH1</sub>	Input high voltage (Ports 1, 3.2, 3.3)		0.8V <sub>CC</sub> +0.3		V <sub>CC</sub> +0.5	V
V <sub>IH2</sub>	Input high voltage (E <sub>A</sub> )		0.2V <sub>CC</sub> +0.9		V <sub>CC</sub> +0.5	V
V <sub>IH3</sub>	Input high voltage (XTAL1, RST)		0.7V <sub>CC</sub>		V <sub>CC</sub> +0.5	V
HYS	Hysteresis (Ports 0, 2, 3, except 3.2, 3.3)		200			mV
HYS1	Hysteresis (Ports 1, 3.2, 3.3)		50			mV
V <sub>OL</sub>	Output voltage low (Ports 1, 2, 3, except 3.1)	I <sub>OL</sub> = 1.6mA			0.45	V
V <sub>OL1</sub>	Output voltage low (Ports 0, ALE, PSEN <sup>1</sup> )	I <sub>OL</sub> = 3.2mA			0.45	V
V <sub>OL2</sub>	Output voltage low P3.1 with bit cleared P3.1 with bit set	I <sub>OL</sub> = 10.0mA I <sub>OL</sub> = 1.6mA			0.50	V
					0.45	V
V <sub>OH</sub>	Output voltage high (Ports 1, 2, 3, except P3.1)	I <sub>OH</sub> = -30µA I <sub>OH</sub> = -10µA	V <sub>CC</sub> -0.7 V <sub>CC</sub> -0.3			V
						V
V <sub>OH1</sub>	Output voltage high (Port 0 in external bus mode, ALE, PSEN)	I <sub>OH</sub> = -3.2mA I <sub>OH</sub> = -200µA	V <sub>CC</sub> -0.7 V <sub>CC</sub> -0.3			V
						V
V <sub>OH2</sub>	Output voltage high P3.1 with bit cleared P3.1 with bit set	I <sub>OH</sub> = -10.0mA I <sub>OH</sub> = -1.6mA	V <sub>CC</sub> -1.5 V <sub>CC</sub> -1.5			V
						V
V <sub>IO</sub>	Offset voltage comparator inputs		-35		+35	mV
V <sub>CR</sub>	Common mode range comparator inputs		0		V <sub>CC</sub>	V
I <sub>IL</sub>	Logical 0 input current (Ports 1, 2, 3, except 3.1)	V <sub>IN</sub> = 0.45V			-75	µA
I <sub>TL</sub>	Logical 1-to-0 transition current (Ports 2, 3, except 3.1, 3.2, 3.3) <sup>4</sup>	See Note 4			-600	µA
I <sub>TL1</sub>	Logical 1-to-0 transition current (Ports 1, 3.2, 3.3)	See Note 4			-450	µA
I <sub>L1</sub>	Input leakage current (Port 0, Port2 in open drain mode) <sup>9</sup>	0.45 < V <sub>IN</sub> < V <sub>CC</sub>	2		40	µA
I <sub>L2</sub>	Input leakage current (E <sub>A</sub> , P3.1)	0.45 < V <sub>IN</sub> < V <sub>CC</sub>	-10		+10	µA
I <sub>LC</sub>	Input leakage current comparator inputs	0 < V <sub>IN</sub> < V <sub>CC</sub>	-1.0		+1.0	µA
I <sub>CC</sub>	Power supply current: <sup>7</sup> Active mode @ 16MHz <sup>5</sup> Idle mode @ 16MHz Power-down mode	See note 6				
				20	30	mA
				8	12	mA
				5	75	µA
R <sub>RST</sub>	Internal reset pull-up resistor	V <sub>IN</sub> = 0V	50		200	kΩ
V <sub>LOW</sub>	Low V <sub>CC</sub> detect voltage		4.0		4.45	V
C <sub>IO</sub>	Pin capacitance <sup>10</sup>	f = 1MHz			10	pF

NOTES: (SEE NEXT PAGE)

## CMOS single-chip 8-bit microcontroller 83C576/87C576

## NOTES TO THE DC ELECTRICAL CHARACTERISTICS TABLE:

- Typical ratings are not guaranteed. The values listed are at room temperature, 5V.
- Capacitive loading on ports 0 and 2 may cause spurious noise to be superimposed on the  $V_{OL}$ s of ALE and ports 1 and 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100pF), the noise pulse on the ALE pin may exceed 0.8V. In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.  $I_{OL}$  can exceed these conditions provided that no single output sinks more than 5mA and no more than two outputs exceed the test conditions.
- Capacitive loading on ports 0 and 2 may cause the  $V_{OH}$  on ALE and  $PSEN$  to momentarily fall below the 0.9V<sub>CC</sub> specification when the address bits are stabilizing.
- Pins of ports 1, 2 and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when  $V_{IN}$  is between  $V_{IH}$  and  $V_{IL}$ .
- $I_{CCMAX}$  at other frequencies can be determined from Figure 30.
- See Figures 30 through 34 for  $I_{CC}$  test conditions.
- Load capacitance for port 0, ALE, and  $PSEN$  = 100pF, load capacitance for all other outputs = 80pF.
- Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:
 

Maximum $I_{OL}$ per port pin:	10mA
Maximum $I_{OL}$ per 8-bit port:	26mA
Maximum total $I_{OL}$ for all outputs:	71mA

If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
- Specification applies to Port 2 when P2OD bit is set.
- 15pF MAX for the EA/ $V_{PP}$  and P0.0 pins.

### A/D CONVERTER DC ELECTRICAL CHARACTERISTICS

$T_{amb} = -40^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			MIN	MAX	
AV <sub>CC</sub>	Analog supply voltage	AV <sub>CC</sub> = V <sub>CC</sub> ± 0.2	4.0	6.0	V
AI <sub>CC</sub>	Analog supply current			1.2	mA
AE	Absolute voltage error			±3	LSB

## CMOS single-chip 8-bit microcontroller 83C576/87C576

## AC ELECTRICAL CHARACTERISTICS

T<sub>amb</sub> = 0°C to +70°C and -40°C to +125°C, V<sub>CC</sub> = 5V ±20%, V<sub>SS</sub> = 0V1, 2

SYMBOL	FIGURE	PARAMETER	VARIABLE CLOCK		UNIT
			MIN	MAX	
1/t <sub>CLCL</sub>	23	Oscillator frequency: Speed Versions 8XC576 E	6	16	MHz
OSCF		Oscillator fail detect frequency	0.6	5.5	MHz
TR		Comparator response time		10	μs
t <sub>LHLL</sub>	23	ALE pulse width	2t <sub>CLCL</sub> -40		ns
t <sub>AVLL</sub>	23	Address valid to ALE low	t <sub>CLCL</sub> -25		ns
t <sub>LLAX</sub>	23	Address hold after ALE low	t <sub>CLCL</sub> -25		ns
t <sub>LLIV</sub>	23	ALE low to valid instruction in		4t <sub>CLCL</sub> -75	ns
t <sub>LLPL</sub>	23	ALE low to PSEN low	t <sub>CLCL</sub> -25		ns
t <sub>PLPH</sub>	23	PSEN pulse width	3t <sub>CLCL</sub> -45		ns
t <sub>PLIV</sub>	23	PSEN low to valid instruction in		3t <sub>CLCL</sub> -70	ns
t <sub>PIXI</sub>	23	Input instruction hold after PSEN	0		ns
t <sub>PIXZ</sub>	23	Input instruction float after PSEN		t <sub>CLCL</sub> -25	ns
t <sub>AVIV</sub>	23	Address to valid instruction in		5t <sub>CLCL</sub> -85	ns
t <sub>PLAZ</sub>	23	PSEN low to address float		10	ns
<b>Data Memory</b>					
t <sub>RLRH</sub>	24, 25	RD pulse width	6t <sub>CLCL</sub> -100		ns
t <sub>WLWH</sub>	24, 25	WR pulse width	6t <sub>CLCL</sub> -100		ns
t <sub>RLDV</sub>	24, 25	RD low to valid data in		5t <sub>CLCL</sub> -110	ns
t <sub>RHDX</sub>	24, 25	Data hold after RD	0		ns
t <sub>RHDZ</sub>	24, 25	Data float after RD		2t <sub>CLCL</sub> -28	ns
t <sub>LLDV</sub>	24, 25	ALE low to valid data in		8t <sub>CLCL</sub> -150	ns
t <sub>AVDV</sub>	24, 25	Address to valid data in		9t <sub>CLCL</sub> -165	ns
t <sub>LLWL</sub>	24, 25	ALE low to RD or WR low	3t <sub>CLCL</sub> -50	3t <sub>CLCL</sub> +50	ns
t <sub>AVWL</sub>	24, 25	Address valid to WR low or RD low	4t <sub>CLCL</sub> -75		ns
t <sub>QVWX</sub>	24, 25	Data valid to WR transition	t <sub>CLCL</sub> -30		ns
t <sub>WHQX</sub>	24, 25	Data hold after WR	t <sub>CLCL</sub> -25		ns
t <sub>RLAZ</sub>	24, 25	RD low to address float		0	ns
t <sub>WHLH</sub>	24, 25	RD or WR high to ALE high	t <sub>CLCL</sub> -25	t <sub>CLCL</sub> +25	ns
<b>External Clock</b>					
t <sub>CHCX</sub>	27	High time	12		ns
t <sub>CLCX</sub>	27	Low time	12		ns
t <sub>CLCH</sub>	27	Rise time		20	ns
t <sub>CHCL</sub>	27	Fall time		20	ns
<b>Shift Register</b>					
t <sub>XLXL</sub>	26	Serial port clock cycle time	12t <sub>CLCL</sub>		ns
t <sub>QVXH</sub>	26	Output data setup to clock rising edge	10t <sub>CLCL</sub> -133		ns
t <sub>XHQX</sub>	26	Output data hold after clock rising edge	2t <sub>CLCL</sub> -60		ns
t <sub>XHDX</sub>	26	Input data hold after clock rising edge	0		ns
t <sub>XHDV</sub>	26	Clock rising edge to input data valid		10t <sub>CLCL</sub> -133	ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- Interfacing the 80C32/52 to devices with float times up to 45ns is permitted. This limited bus contention will not cause damage to Port 0 drivers.

## CMOS single-chip 8-bit microcontroller 83C576/87C576

## UPI AC ELECTRICAL CHARACTERISTICS

 $T_{amb} = -40^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$ ,  $AV_{CC} = V_{CC} = 5\text{V} \pm 20\%$ 

SYMBOL	PARAMETER	MIN	MAX	UNIT
$t_{AR}$	CS#, A setup to RD#	0		ns
$t_{RA}$	CS#, A hold after RD#	0		ns
$t_{RR}$	RD# pulse width	60		ns
$t_{AD}$	CS#, A to data out delay		70	ns
$t_{RD}$	RD# to data out delay		60	ns
$t_{DF}$	RD# to data float delay		5	ns
$t_{AW}$	CS#, A setup to WR#	0		ns
$t_{WA}$	CS#, A hold after WR#	0		ns
$t_{WW}$	WR# pulse width	60		ns
$t_{DW}$	Data setup to WR#	60		ns
$t_{WD}$	Data hold after WR#	0		ns

## EXPLANATION OF THE AC SYMBOLS

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

A – Address

C – Clock

D – Input data

H – Logic level high

I – Instruction (program memory contents)

L – Logic level low, or ALE

P – PSEN

Q – Output data

R – RD signal

t – Time

V – Valid

W – WR signal

X – No longer a valid logic level

Z – Float

Examples:  $t_{AVLL}$  = Time for address valid to ALE low.

$t_{LLPL}$  = Time for ALE low to PSEN low.

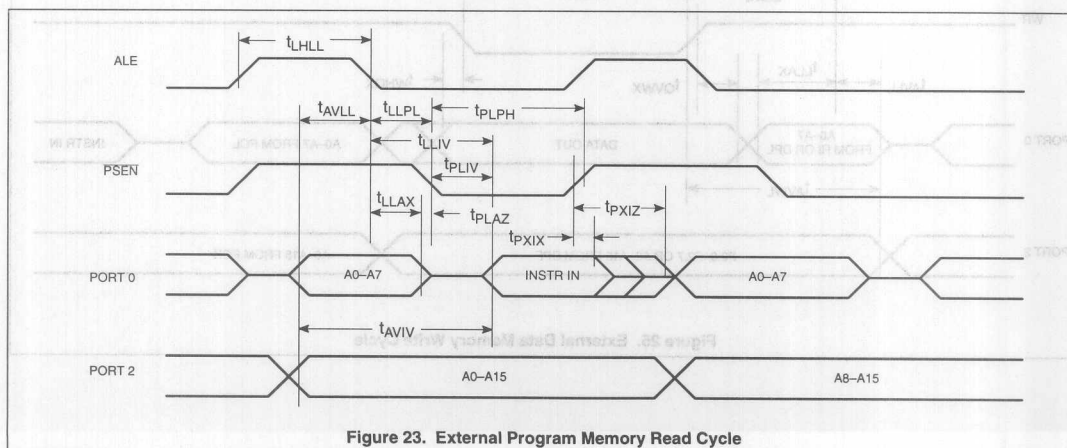
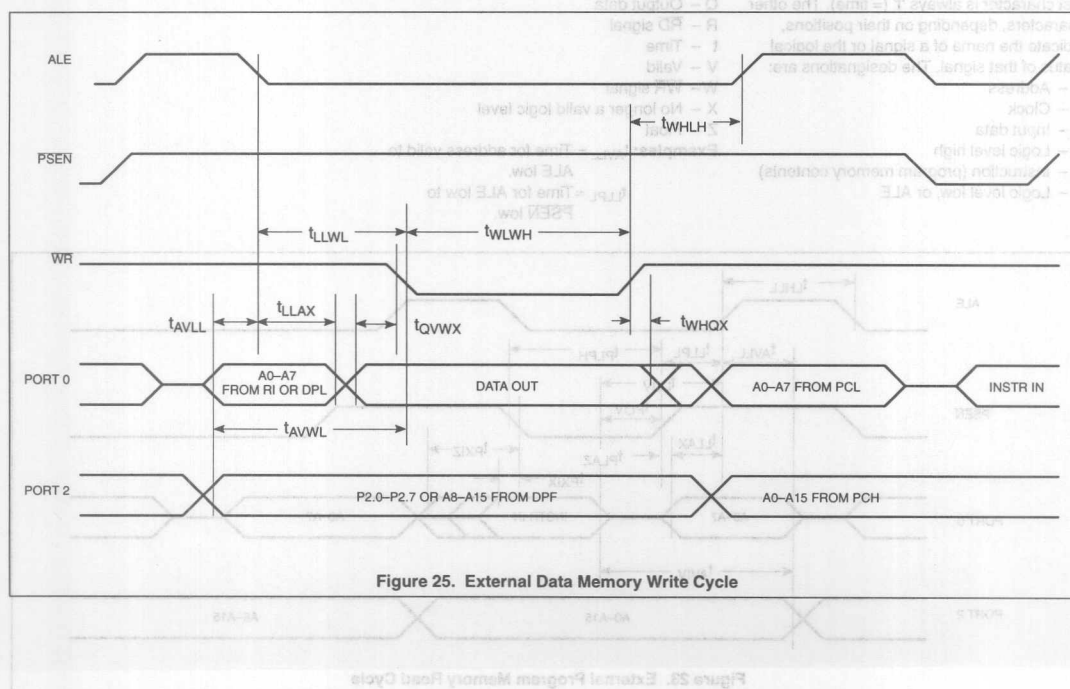
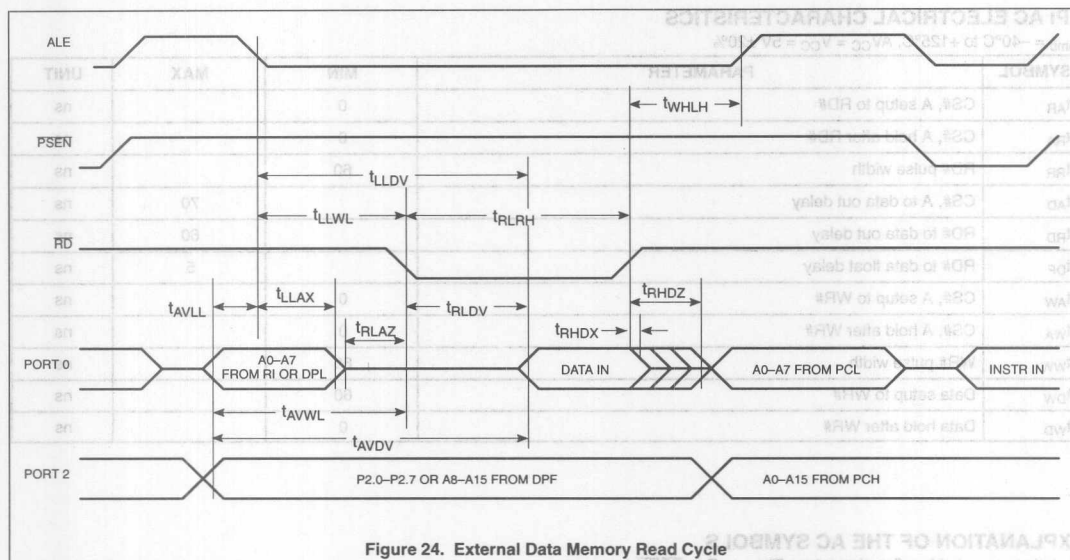


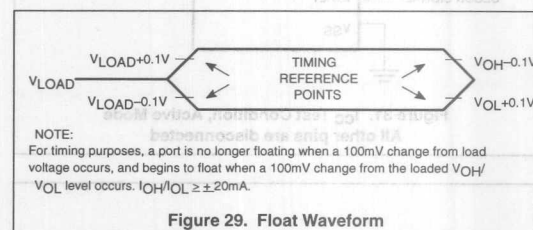
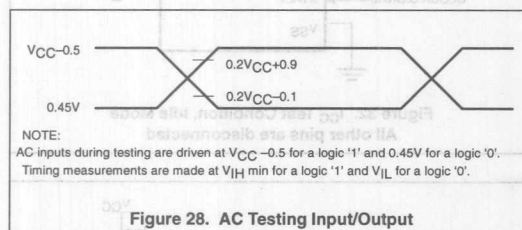
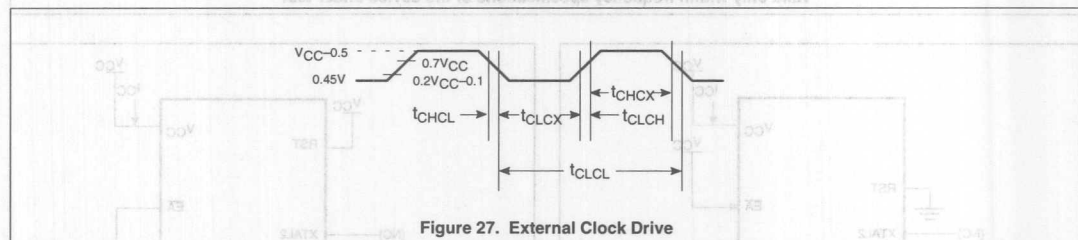
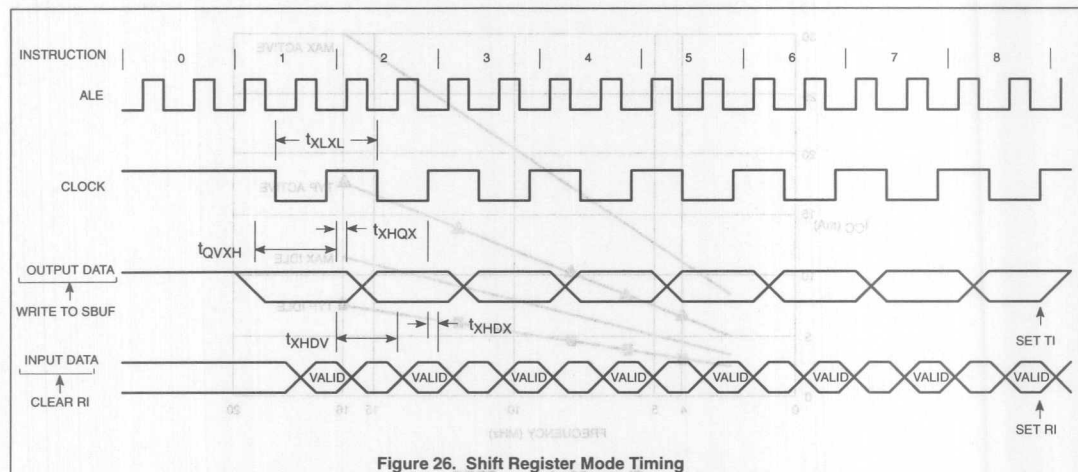
Figure 23. External Program Memory Read Cycle

## CMOS single-chip 8-bit microcontroller 83C576/87C576





## CMOS single-chip 8-bit microcontroller 83C576/87C576



## CMOS single-chip 8-bit microcontroller

83C576/87C576

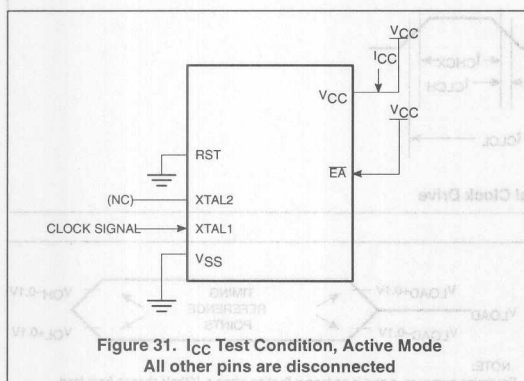
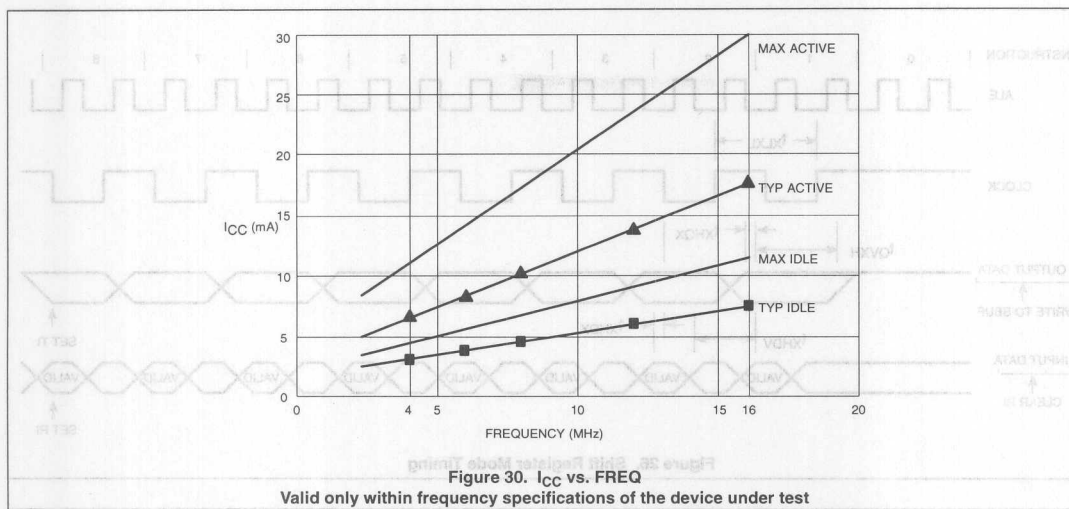


Figure 31.  $I_{CC}$  Test Condition, Active Mode  
All other pins are disconnected

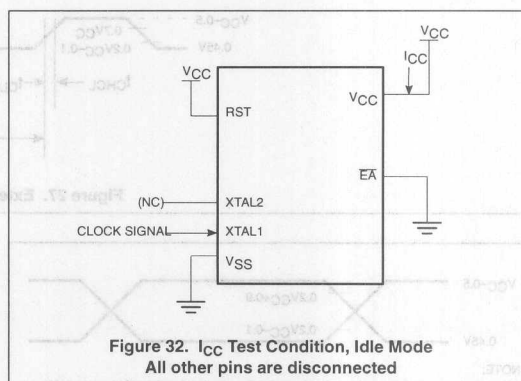


Figure 32.  $I_{CC}$  Test Condition, Idle Mode  
All other pins are disconnected

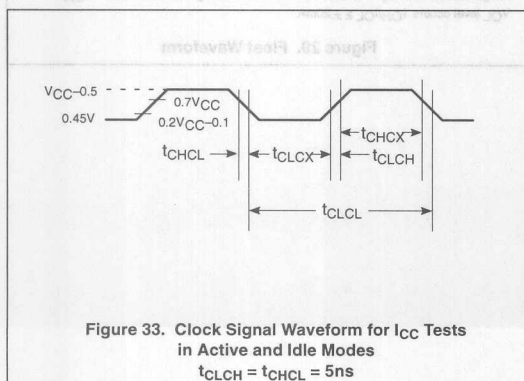


Figure 33. Clock Signal Waveform for  $I_{CC}$  Tests  
in Active and Idle Modes  
 $t_{CLCH} = t_{CHCL} = 5ns$

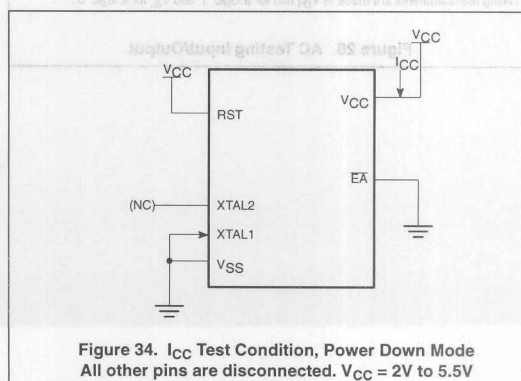


Figure 34.  $I_{CC}$  Test Condition, Power Down Mode  
All other pins are disconnected.  $V_{CC} = 2V$  to  $5.5V$

## CMOS single-chip 8-bit microcontroller 83C576/87C576

### EPROM CHARACTERISTICS

To put the 87C576 in the EPROM programming mode, PSEN must be held high during power up, then driven low with reset active. The 87C576 is programmed by using a modified Quick-Pulse Programming™ algorithm. It differs from older methods in the value used for V<sub>PP</sub> (programming supply voltage) and in the width and number of the ALE/PROG pulses.

The 87C576 contains two signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes identify the device as an 87C576 manufactured by Philips.

Table 3 shows the logic levels for reading the signature byte, and for programming the program memory, the encryption table, and the security bits. The circuit configuration and waveforms for quick-pulse programming are shown in Figures 35 and 36. Figure 37 shows the circuit configuration for normal program memory verification.

### On-Board Programming (OBP)

The On-Board Programming facility consists of a series of internal hardware resources coupled with internal firmware to facilitate remote programming of the 87C576 through the serial port.

The OBP function is invoked by having the EA/V<sub>PP</sub> pin at the V<sub>PP</sub> voltage level at the time that the part exits reset. The OBP function only requires that the Tx<sub>D</sub>, Rx<sub>D</sub>, V<sub>SS</sub>, V<sub>CC</sub>, and V<sub>PP</sub> pins be connected to an external circuit in order to use this feature.

The OBP feature provides for the use of a wide range of baud rates independent of the oscillator frequency used. It is also adaptable to a wide range of oscillator frequencies. The OBP facility provides for both auto-echo and no-echo of received characters. The OBP feature requires that an initial character, either an uppercase U or a 5 be sent to the 87C576 to establish the baud rate to be used. This character also indicates whether an auto-echo (a U) or no-echo (a 5) is to be used.

Once baud rate initialization has been performed, the OBP facility only accepts Intel Hex records. The record-type field of these hex records are used to indicate either commands or data for the OBP facility. The maximum number of data bytes in a record is

limited to 16 (decimal). These commands/data are summarized below:

Record Type	Command/Data Function
00	Data record, programs the part with data indicated in record starting with load address in the record
01	EOF record, no operation
02	Specify timing parameters <ul style="list-style-type: none"> <li>– rec length = 3 bytes</li> <li>– load address = 0000</li> <li>– 1st byte = timer count for 50µs programming pulse</li> <li>– 2nd byte = timer count for 10µs delay between pulses</li> <li>– 3rd byte = 0AH</li> </ul>
03	Program security bits <ul style="list-style-type: none"> <li>– rec length = 1 byte</li> <li>– load address = 0000</li> <li>– 1st byte = sec bit values (xxxx xxB2B1)</li> </ul>
04	Display contents of USER EPROM array <ul style="list-style-type: none"> <li>– rec length = 00</li> <li>– load address = 0000</li> </ul>
05	Verify security bit status <ul style="list-style-type: none"> <li>– rec length = 00</li> <li>– load address = 0000</li> </ul>
06	Pulse timing test, pulses P3.2 for timing verification <ul style="list-style-type: none"> <li>– rec length = 00</li> <li>– load address = 0000</li> </ul>

### Quick-Pulse Programming

The setup for microcontroller quick-pulse programming is shown in Figure 35. Note that the 87C576 is running with a 4 to 6MHz oscillator. The reason the oscillator needs to be running is that the device is executing internal address and program data transfers.

The address of the EPROM location to be programmed is applied to ports 1 and 2, as shown in Figure 35. The code byte to be programmed into that location is applied to port 0. RST, PSEN and pins of ports 2 and 3 specified in Table 3 are held at the 'Program Code Data' levels indicated in Table 3. The ALE/PROG is pulsed low 25 times as shown in Figure 36.

To program the encryption table, repeat the 25 pulse programming sequence for addresses 0 through 1FH, using the 'Pgm

Encryption Table' levels. Do not forget that after the encryption table is programmed, verification cycles will produce only encrypted data.

To program the security bits, repeat the 25 pulse programming sequence using the 'Pgm Security Bit' levels. After one security bit is programmed, further programming of the code memory and encryption table is disabled. However, the other security bit can still be programmed.

Note that the EA/V<sub>PP</sub> pin must not be allowed to go above the maximum specified V<sub>PP</sub> level for any amount of time. Even a narrow glitch above that voltage can cause permanent damage to the device. The V<sub>PP</sub> source should be well regulated and free of glitches and overshoot.

### Program Verification

If security bit 2 has not been programmed, the on-chip program memory can be read out for program verification. The address of the program memory locations to be read is applied to ports 1 and 2 as shown in Figure 37. The other pins are held at the 'Verify Code Data' levels indicated in Table 3. The contents of the address location will be emitted on port 0. External pull-ups are required on port 0 for this operation.

If the encryption table has been programmed, the data presented at port 0 will be the exclusive NOR of the program byte with one of the encryption bytes. The user will have to know the encryption table contents in order to correctly decode the verification data. The encryption table itself cannot be read out.

### Reading the Signature Bytes

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. The values are:

(030H) = 15H indicates manufactured by Philips  
(B0H) = B6H indicates 87C576

### Program/Verify Algorithms

Any algorithm in agreement with the conditions listed in Table 3, and which satisfies the timing specifications, is suitable.

™Trademark phrase of Intel Corporation.

## CMOS single-chip 8-bit microcontroller 83C576/87C576

## Erasure Characteristics

Erasure of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent

erasure. **For this and secondary effects, it is recommended that an opaque label be placed over the window.** For elevated temperature or environments where solvents are being used, apply Kapton tape Fluorglas part number 2345-5, or equivalent.

The recommended erasure procedure is exposure to ultraviolet light (at 2537

angstroms) to an integrated dose of at least 15W-s/cm<sup>2</sup>. Exposing the EPROM to an ultraviolet lamp of 12,000µW/cm<sup>2</sup> rating for 20 to 39 minutes, at a distance of about 1 inch, should be sufficient.

Erasure leaves the array in an all 1s state.

Table 3. EPROM Programming Modes

MODE	RST	PSEN	ALE/PROG	EA/V <sub>PP</sub>	P2.7	P2.6	P3.7	P3.6
Read signature	0	0	1	1	0	0	0	0
Program code data	0	0	0	V <sub>PP</sub>	1	0	1	1
Verify code data	0	0	1	1	0	0	1	1
Pgm encryption table	0	0	0	V <sub>PP</sub>	1	0	1	0
Pgm security bit 1	0	0	0	V <sub>PP</sub>	1	1	1	1
Pgm security bit 2	0	0	0	V <sub>PP</sub>	1	1	0	0

## NOTES:

- '0' = Valid low for that pin, '1' = valid high for that pin.
- V<sub>PP</sub> = 12.75V ±0.25V.
- V<sub>CC</sub> = 5V ±10% during programming and verification.

\* ALE/PROG receives 25 programming pulses while V<sub>PP</sub> is held at 12.75V. Each programming pulse is low for 100µs (±10µs) and high for a minimum of 10µs.

If the encryption table has been programmed, the data presented at port 0 will be the exclusive NOR of the program byte with one of the encryption bytes. The user will have to know the encryption table contents in order to correctly decode the verification data. The encryption table itself cannot be read out.

Reading the Signature Bytes  
The signature bytes are read by the same procedure as a normal verification of locations 000H and 001H, except that P2.6 and P2.7 need to be pulled to a logic low. The values are:

(000H) = 15H indicates manufactured by Philips  
(001H) = 80H indicates 87C576

**Program/Verify Algorithms**  
Any algorithm in agreement with the conditions listed in Table 2, and which satisfies the timing specifications, is suitable.

**Quick-Pulse Programming**  
The setup for microcontroller quick-pulse programming is shown in Figure 36. Note that the 87C576 is running with a 4 to 6MHz oscillator. The reason the oscillator needs to be running is that the device is executing internal address and program data transfers. The address of the EPROM location to be programmed is applied to ports 1 and 2 as shown in Figure 36. The code byte to be programmed into that location is applied to port 0. RST, PSEN and pins of ports 5 and 3 specified in Table 3 are held in the Program Code Data levels indicated in Table 2. The ALE/PROG is pulsed low 25 times as shown in Figure 36.

To program the encryption table, repeat the 25 pulse programming sequence for addresses 0 through 1FH, using the "Pgm

The OBP function is invoked by having the EA/V<sub>PP</sub> pin at the V<sub>PP</sub> voltage level at the time that the next reset, the OBP function only requires that the TXD, RXD, V<sub>CC</sub>, and V<sub>PP</sub> pins be connected to an external circuit in order to use this feature.

The OBP feature provides for the use of a wide range of baud rates independent of the oscillator frequency used. It is also adaptable to wide ranges of oscillator frequencies. The OBP facility provides for both auto-echo and no-echo of received characters. The OBP feature requires that an initial character either an uppercase U or a no-echo (N) be used. This character also indicates whether an auto-echo (a U) or no-echo (a N) is to be used.

Once baud rate initialization has been performed, the OBP facility only accepts initial hex records. The record-type field of these hex records are used to indicate either commands or data for the OBP facility. The maximum number of data bytes in a record is

## CMOS single-chip 8-bit microcontroller 83C576/87C576

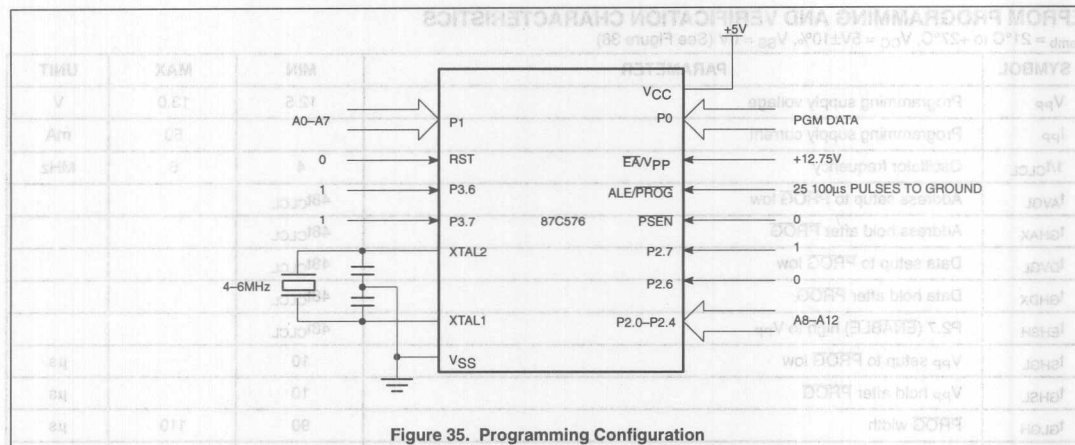


Figure 35. Programming Configuration

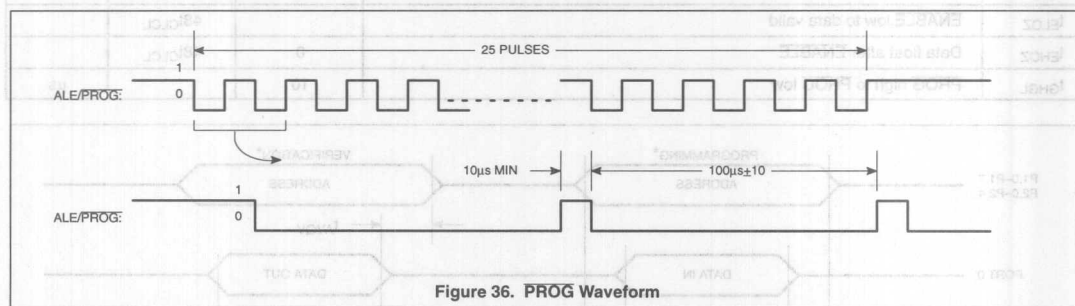


Figure 36.  $\overline{\text{PROG}}$  Waveform

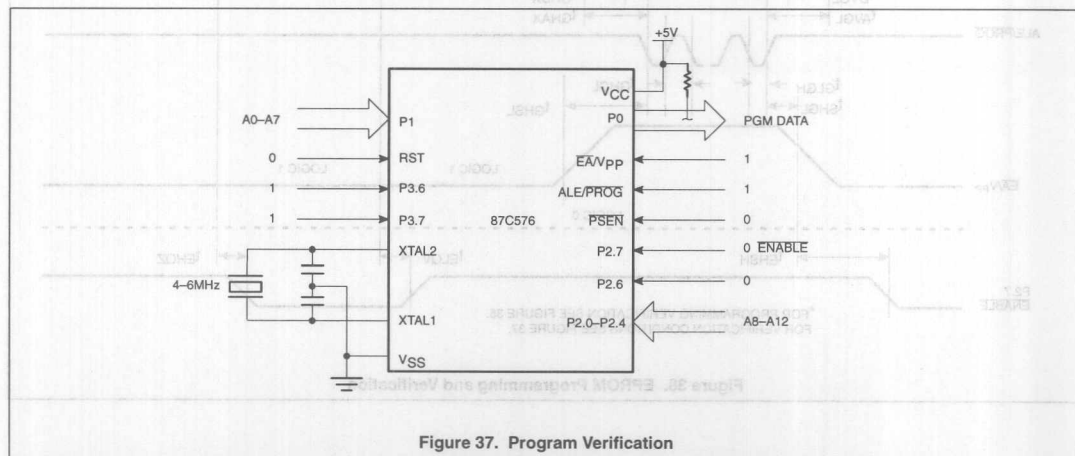


Figure 37. Program Verification



## CMOS single-chip 8-bit microcontroller 83C576/87C576

## EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

$T_{amb} = 21^{\circ}\text{C}$  to  $+27^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$  (See Figure 38)

SYMBOL	PARAMETER	MIN	MAX	UNIT
$V_{PP}$	Programming supply voltage	12.5	13.0	V
$I_{PP}$	Programming supply current		50	mA
$1/t_{CLCL}$	Oscillator frequency	4	6	MHz
$t_{AVGL}$	Address setup to PROG low	$48t_{CLCL}$		
$t_{GHAX}$	Address hold after PROG	$48t_{CLCL}$		
$t_{DVGL}$	Data setup to PROG low	$48t_{CLCL}$		
$t_{GHDX}$	Data hold after PROG	$48t_{CLCL}$		
$t_{EHS}$	P2.7 (ENABLE) high to $V_{PP}$	$48t_{CLCL}$		
$t_{SHGL}$	$V_{PP}$ setup to PROG low	10		$\mu\text{s}$
$t_{GHSL}$	$V_{PP}$ hold after PROG	10		$\mu\text{s}$
$t_{GLGH}$	PROG width	90	110	$\mu\text{s}$
$t_{AVQV}$	Address to data valid		$48t_{CLCL}$	
$t_{ELQZ}$	ENABLE low to data valid		$48t_{CLCL}$	
$t_{EHQZ}$	Data float after ENABLE	0	$48t_{CLCL}$	
$t_{GHGL}$	PROG high to PROG low	10		$\mu\text{s}$

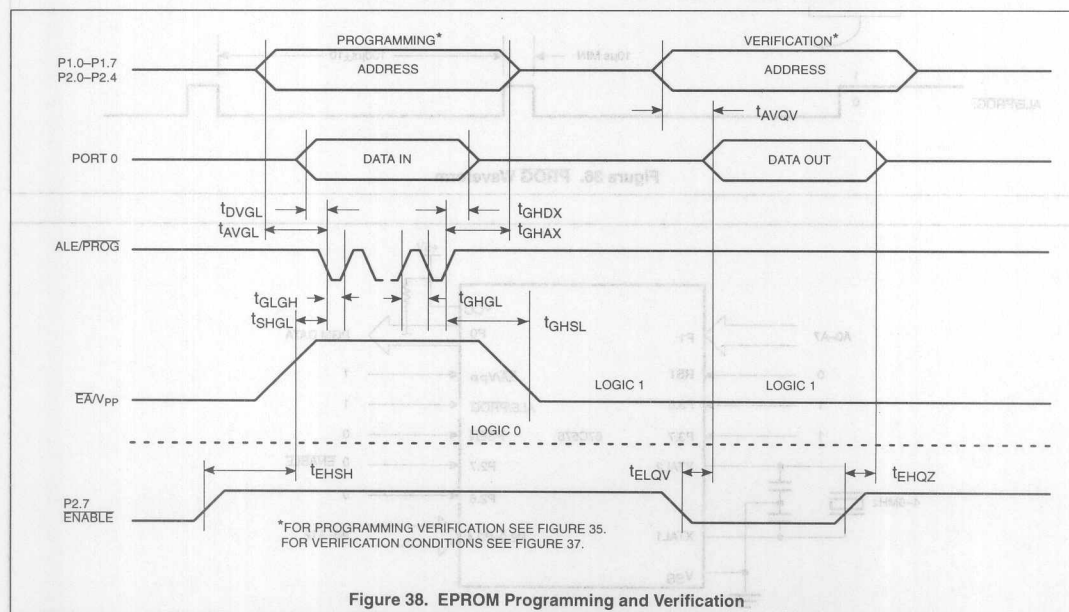


Figure 38. EPROM Programming and Verification

## Low-voltage single-chip 8-bit microcontroller

## 80CL580/83CL580

## GENERAL DESCRIPTION

The 83CL580 is manufactured in an advanced CMOS technology. The instruction set of the 83CL580 is based on that of the 8051. The 83CL580 is an 8-bit general purpose microcontroller especially suited for cordless telephones and mobile communication applications. The device has low power consumption and a wide range of supply voltages. For emulation purposes, the 85CL580 (Piggy-back version) with 256 bytes of RAM is recommended. The 83CL580 has two software selectable modes of reduced activity for further power reduction: Idle and Power-down. The 83CL580 also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of over 100 instructions: 49 one-byte, 46 two-byte, and 16 three-byte.

## FEATURES

- Full static 80C51 CPU
- 8-bit CPU, ROM, RAM, I/O in a single 56-lead VSO56 or 64-lead QFP64 package
- 6K × 8 ROM, expandable externally to 64K bytes
- 256 bytes RAM, expandable externally to 64K bytes
- Five 8-bit ports, 40 I/O lines
- Three 16-bit timer / event counters
- External memory expandable up to 128K, external ROM up to 64K and / or RAM up to 64K
- On-chip oscillator suitable for RC, LC, quartz crystal or ceramic resonator
- Fifteen source, fifteen vector interrupt structure with two priority levels
- Full duplex serial UART
- I<sup>2</sup>C-bus interface for serial transfer on two lines
- A/D converter with power-down mode (8-bit, 4 inputs)
- Pulse width modulated output (8-bit resolution)
- Watchdog timer
- Enhanced architecture with:
  - non-page oriented instructions
  - four eight byte RAM register banks
  - direct addressing
  - multiply, divide, subtract and compare instructions
  - stack depth limited only by available internal RAM (max. 256 bytes)
- Power-Down and IDLE instructions
- Wake-up via external interrupts at Port 1
- Single supply voltage of 2.5V to 6.0V
- Frequency range of 0 to 12MHz
- Very low current consumption: typically 4.5mA at 2.5V / 8MHz
- Operating temperature range: -40 to +85°C

## ORDERING INFORMATION

PHILIPS PART ORDER NUMBER PART MARKING		PHILIPS NORTH AMERICA <sup>1</sup> PART ORDER NUMBER		TEMPERATURE RANGE °C AND PACKAGE	FREQUENCY	DRAWING NUMBER
ROMless	ROM	ROMless	ROM			
P80CL580HFT	P83CL580HFT	P80CL580HF D	P83CL580HF D	-40 to +85 56-Lead Plastic VSO (Very Small Outline) Dual In-line Package	32KHz to 12MHz	SOT190
P80CL580HFH	P83CL580HFH	P80CL580HF B	P83CL580HF B	-40 to +85 64-Lead Plastic Quad Flat Pack	32KHz to 12MHz	SOT319

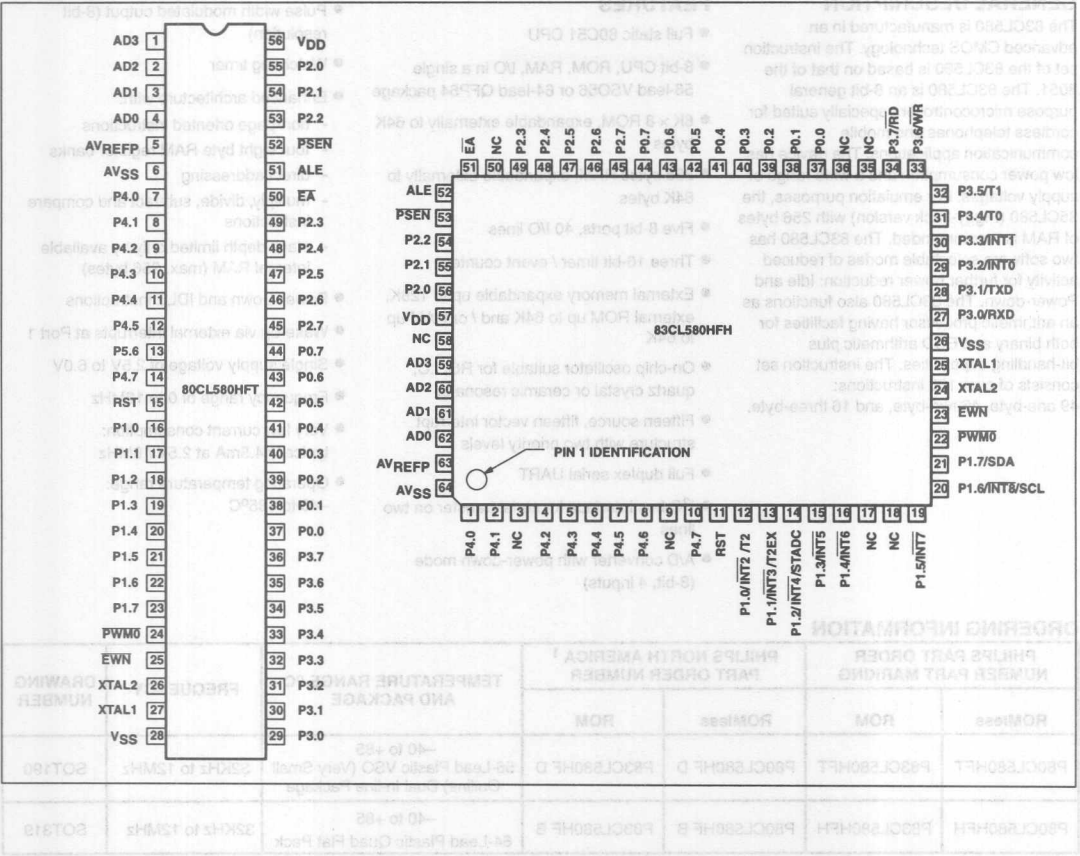
## NOTE:

1. Parts ordered by the Philips North America part number will be marked with the Philips part marking.

Low-voltage single-chip 8-bit microcontroller

80CL580/83CL580

PIN CONFIGURATIONS



## Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

## PIN DESCRIPTIONS

PIN		DESIGNATION	FUNCTION
VSO56	QFP64		
16	12	P1.0/INT2/T2	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pullups. Port 1 pins that have 1s written to them are pulled HIGH by the internal pullups, and in that state can be used as inputs. The Port 1 output buffer can sink/source 4 LS TTL loads. As inputs, Port 1 pins that are externally pulled LOW will source current ( $I_{IL}$ in the characteristics) due to the internal pullups. Port 1 also serves the alternative functions INT2 to INT8, Timer T2 external input external trigger of A/D conversion, and for the I <sup>2</sup> C-bus interface.
17	13	P1.1/INT3/T2EX	
18	14	P1.2/INT4/STADC	
19	15	P1.3/INT5	
20	16	P1.4/INT6	
21	19	P1.5/INT7	
22	20	P1.6/INT8/SCL	
23	21	P1.7/SDA	
15	11	RST	<b>Reset:</b> A high level on this pin for two machine cycles while the oscillator is running resets the device.
29 - 36	27 - 34	P3.0 - P3.7	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 3 output buffers can sink/source 4 LS TTL inputs. Port 3 pins that have 1s written to them are pulled HIGH by the internal pull-ups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally pulled LOW will source current ( $I_{IL}$ in the characteristics) due to the internal pull-ups.
29	27	P3.0/RXD/data	
30	28	P3.1/TxD/clock	<b>RXD/data:</b> UART serial port receiver data input (asynchronous) or data input/output (synchronous). <b>TxD/clock:</b> UART serial port transmitter data output (asynchronous) or clock output (synchronous).
31	29	P3.2/INT0	<b>INT0:</b> external interrupt 0.
32	30	P3.3/INT1	<b>INT1:</b> external interrupt 1.
33	31	P3.4/T0	<b>T0:</b> Timer 0 external input.
34	32	P3.5/T1	<b>T1:</b> Timer 1 external input.
35	33	P3.6/WR	<b>WR:</b> external data memory write strobe.
36	34	P3.7/RD	<b>RD:</b> external data memory read strobe.
6	64	AV <sub>SS</sub>	Analogue ground
5	63	AV <sub>REFP</sub>	High-end of analogue to digital conversion reference resistor
1 - 4	59 - 62	AD3-AD0	Four input channels to ADC
24	22	PWM0	Pulse width modulation output 0
25	23	EWN	Enable watchdog timer
26	24	XTAL2	<b>Crystal output:</b> Output of the inverting amplifier of the oscillator. Left open when external clock is used.
27	25	XTAL1	<b>Crystal input:</b> Input to the inverting amplifier of the oscillator, also the input for an externally generated clock source.
28	26	V <sub>SS</sub>	<b>Ground:</b> circuit ground potential.
55 - 53	56 - 54	P2.0 - P2.2	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pullups. Port 2 pins that have 1s written to them are pulled HIGH by the internal pullups, and in that state can be used as inputs. The Port 2 output buffer can sink/source 4 LS TTL loads. Port 2 emits the high-order address byte during accesses to external memory that use 16-bit addresses (MOVX @DPTR). In this application it uses the strong internal pullups when emitting 1s. During accesses to external memory that use 8-bit addresses (MOVX @Ri). Port 2 emits the contents of the P2 Special Function Register.
49 - 45	49 - 45	P2.3 - P2.7	
52	53	PSEN	<b>Program store enable output:</b> Read strobe to external program memory. When executing code out of external program memory, PSEN is activated twice each machine cycle. However, during each access to external data memory two PSEN activations are skipped.
51	52	ALE	<b>Address Latch Enable:</b> Output pulse for latching the low byte of the address during access to external memory. ALE is emitted at a constant rate of 1/6 of the oscillator frequency, and may be used for external timing or clocking purposes
50	51	EA	<b>External Access:</b> When EA is held High the CPU executes out of internal program memory (unless the program counter exceeds 17FFH). Holding EA LOW forces the CPU to execute out of external memory regardless of the value of the program counter
37 - 44	37 - 44	P0.0 - P0.7	<b>Port 0:</b> Port 0 is an 8-bit open drain bidirectional I/O port. As an open drain output port it can sink 8 LS TTL loads. Port 0 pins that have 1s written to them float, and in that state will function as high impedance inputs. Port 0 is also the multiplexed low order address and data bus during access to external memory. In this application it uses strong internal pull-ups when emitting logic 1s
7-14	1 - 2, 4 - 8, 10	P4.0 - P4.7	<b>Port 4:</b> 8-bit bidirectional I/O port.
56	57	V <sub>DD</sub>	Power supply.

# Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

## 1.0 FUNCTIONAL DESCRIPTION

### General

The 83CL580 is a stand-alone high-performance CMOS microcontroller designed for use in real-time applications such as cordless telephone and mobile communications, instrumentation, industrial control, intelligent computer peripherals and consumer products.

The device provides hardware features, architectural enhancements and new instructions to function as a controller for applications requiring up to 64K bytes of program memory and/or up to 64K bytes of data storage.

The 83CL580 contains a non-volatile 6K byte x 8 read-only program memory; a static 256 byte x 8 read/write data memory; 40 I/O lines; three 16-bit timer/event counters; a fifteen-source two priority-level, nested interrupt structure and on-chip oscillator and timing circuit, 4-channel 8-bit A/D converter, watchdog timer, and pulse width modulation output.

The device has two software selectable modes of reduced activity for power reduction; IDLE and Power-down. The Idle mode freezes the CPU while allowing the RAM, timers, serial I/O and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes

the oscillator causing all other chip functions to be inoperative.

In addition, the device provides an I<sup>2</sup>C-bus serial I/O port with byte oriented master and slave functions, which allows communication with the whole family of I<sup>2</sup>C-bus compatible ICs and a standard UART serial interface.

### CPU Timing

A machine cycle consists of a sequence of 6 states. Each state time lasts for two oscillator periods, thus a machine cycle takes 12 oscillator periods or 1  $\mu$ s if the oscillator frequency is 12 MHz.

### 1.1 Memory Organization

The 83CL580 has a 6K Program Memory (ROM) plus 256 bytes of Data Memory (RAM) on board. The device has separate address spaces for Program and Data Memory (see Figure 1). Using Ports P0 and P2, the 83CL580 can address up to 64K bytes of external memory. The CPU generates both read and write signals ( $\overline{RD}$  and  $\overline{WR}$ ) for external Data Memory accesses, and the read strobe ( $\overline{PSEN}$ ) for external Program Memory.

#### 1.1.1 Program Memory

The 83CL580 contains 6K bytes of internal ROM. After reset the CPU begins execution at location 0000H. The lower 6K bytes of Program Memory can be implemented in

either on-chip ROM or external Memory. If the EA pin is strapped to  $V_{DD}$ , then program memory fetches from addresses 0000H through 17FFFH are directed to the internal ROM. Fetches from addresses 18000H through FFFFH are directed to external ROM. Program counter values greater than 17FFFH are automatically addressed to external memory regardless of the state of the EA pin.

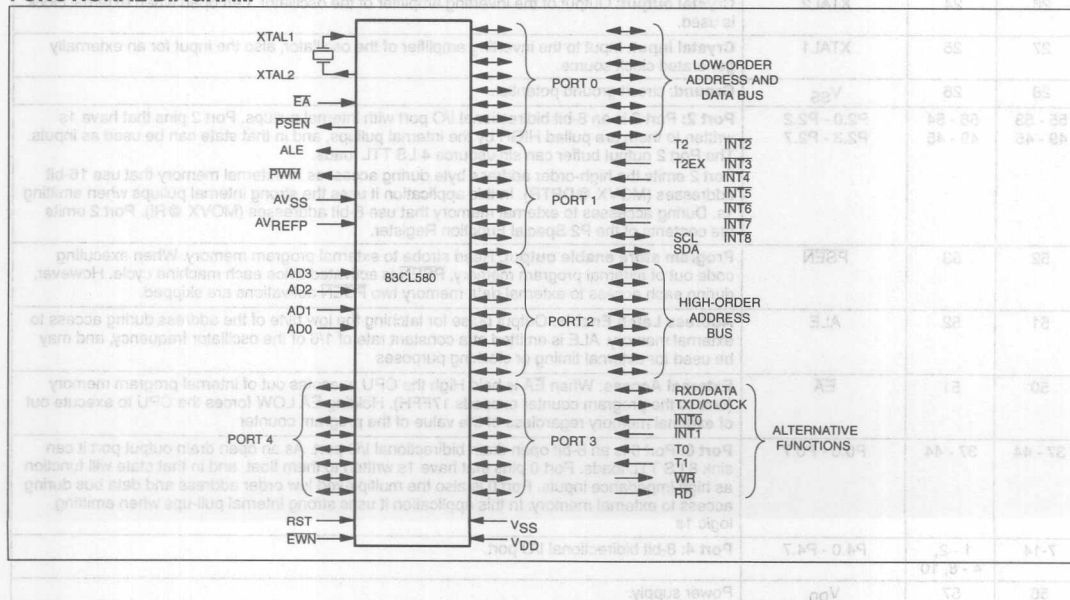
#### 1.1.2 Data Memory

The 83CL580 contains 256 bytes of internal RAM and 40 Special Function Registers (SFR). Figure 1 shows the internal Data Memory space divided into the Lower 128, the Upper 128, and the SFR space. Internal RAM locations 0-127 are directly and indirectly addressable. Internal RAM locations 128-255 are only indirectly addressable. The special function register locations 128-255 are only directly addressable.

#### 1.1.3 Special Function Registers

The upper 128 bytes are the address locations of the SFRs. Figure 3 shows the Special Function Register (SFR) space. SFRs include the port latches, timers, etc. peripheral control, serial I/O registers, etc. These registers can only be accessed by direct addressing. There are 128 bit addressable locations in the SFR address space (SFRs with addresses divisible by eight).

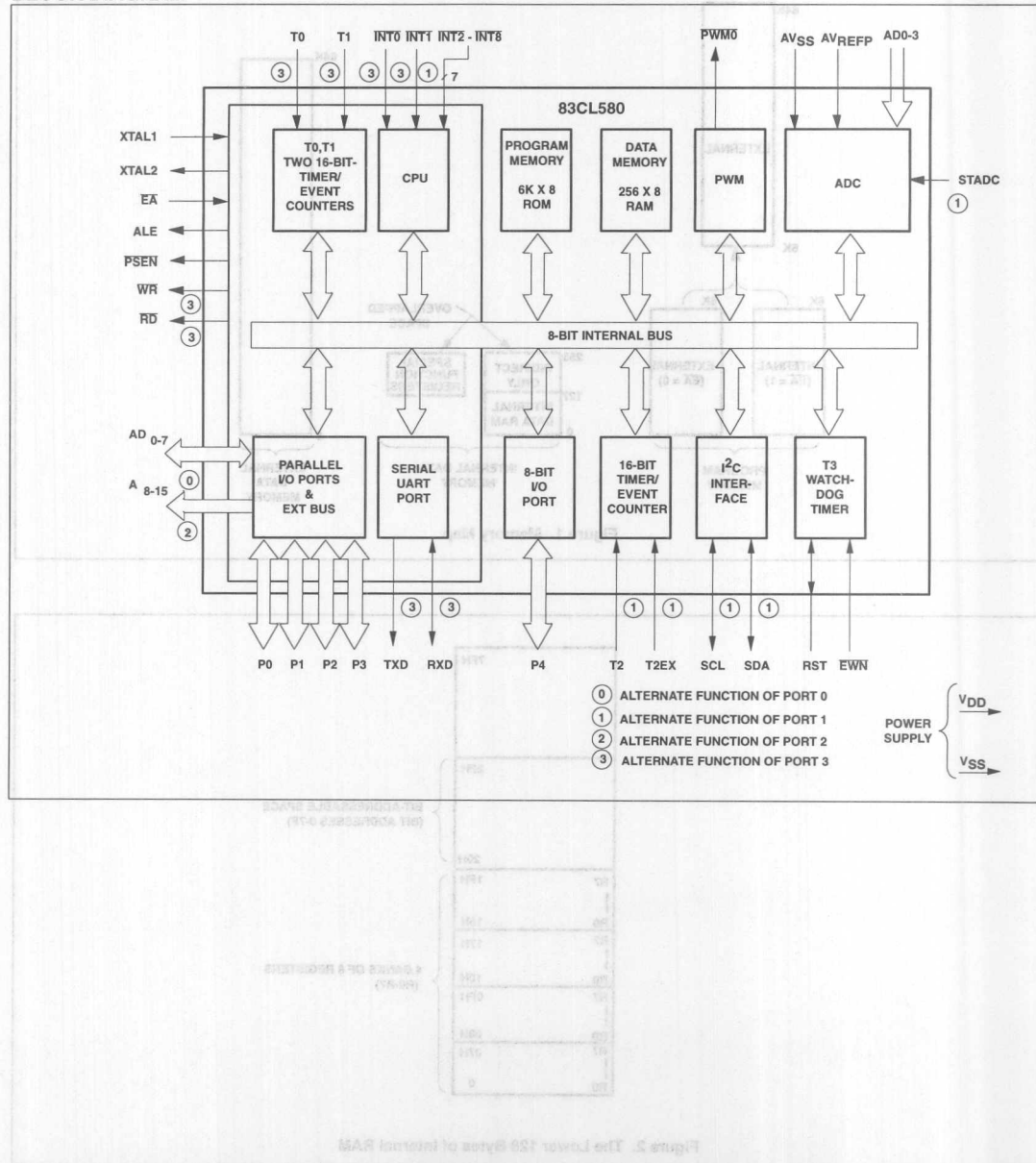
## FUNCTIONAL DIAGRAM



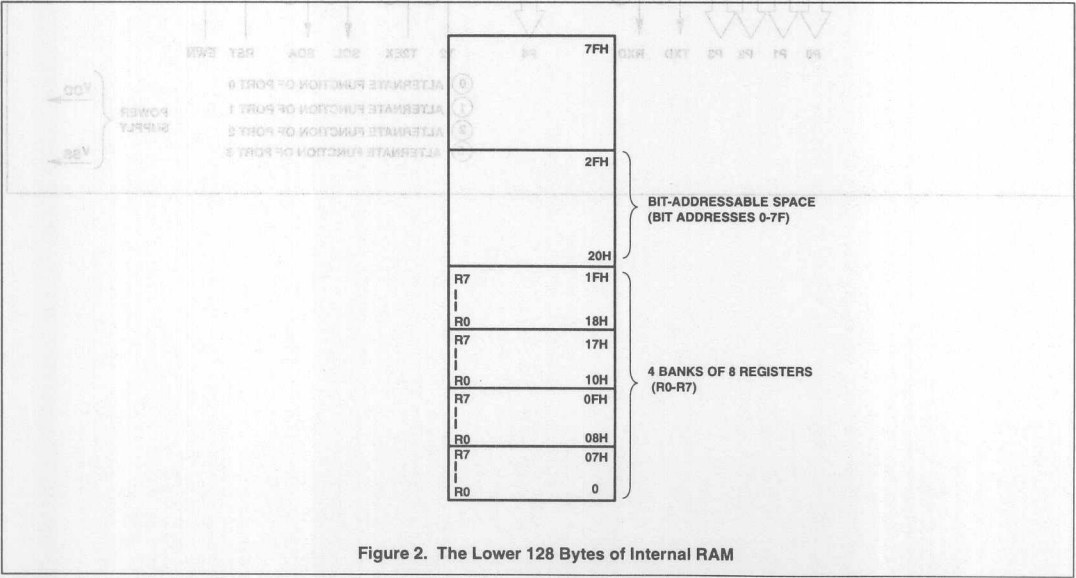
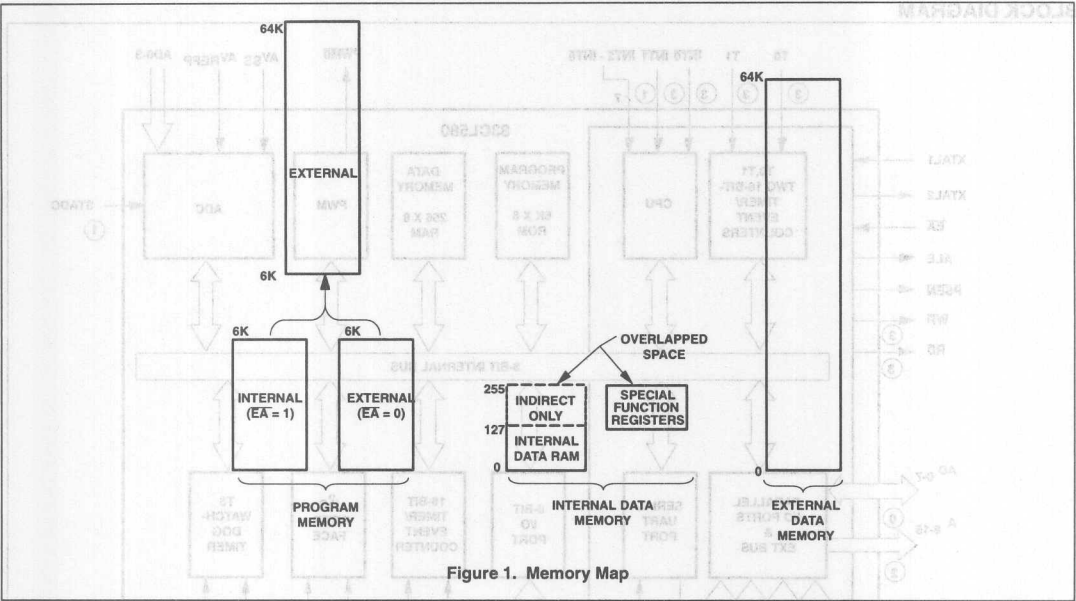


## Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

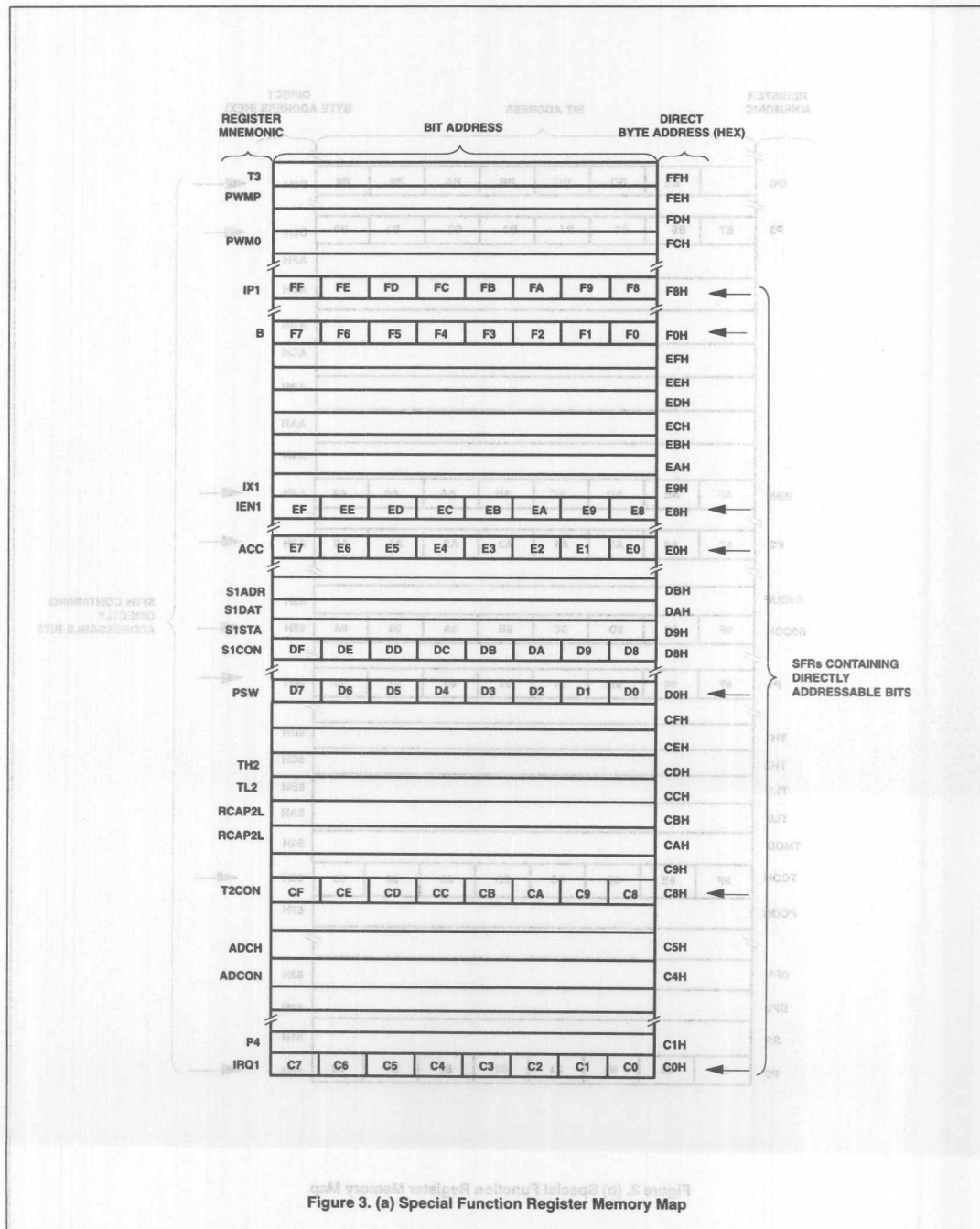
## BLOCK DIAGRAM



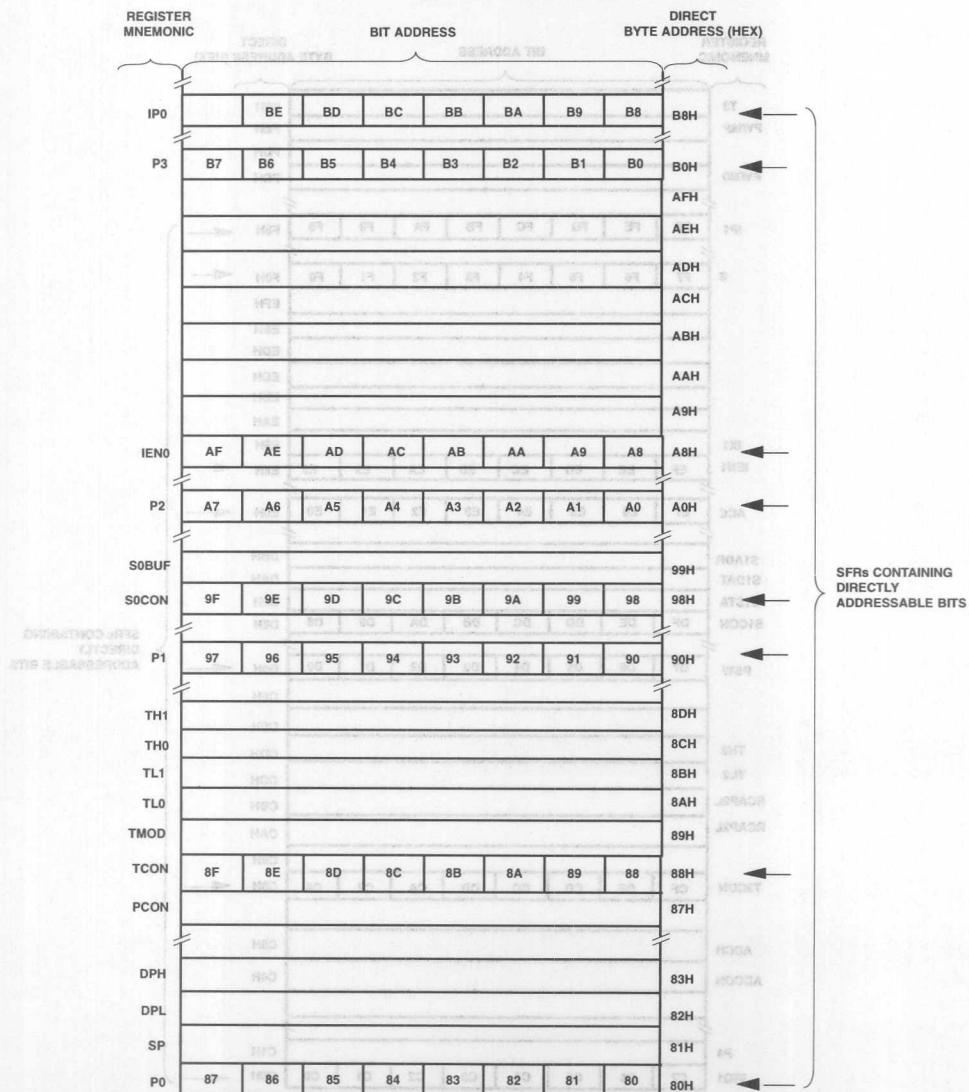
Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580



## Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580



Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580



**Figure 3. (b) Special Function Register Memory Map**

## Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

### 1.1.4 Addressing

The 83CL580 has five methods for addressing source operands:

- Register
- Direct
- Register-Indirect
- Immediate
- Base-Register-plus Index-Register-indirect

The first three methods can be used for addressing destination operands. Most instructions have a "destination/source" field that specifies data type, addressing methods, and operands involved. For operations other than MOVs, the destination operand is also a source operand.

Access to memory addressing is as follows:

- Registers in one of the four register banks through register, direct or indirect.
- Internal RAM (256 bytes) through direct or register-indirect.
- Special Function Register through Direct.
- External data memory through Register-Indirect
- Program memory look-up tables through Base-Register-plus index-Register-indirect.

### 1.2 I/O facilities

#### 1.2.1 Ports

The 83CL580 has 40 I/O lines treated as one port plus 32 individually addressable bits or

as five parallel 8-bit addressable ports.

Port 0, 1, 2 and 3 perform the following alternate functions:

- Port 0: provides the multiplexed low-order address and data bus for expanding the device with standard memories and peripherals.
- Port 1:
  - (1) provides the inputs for the external interrupts INT2 / INT8
  - (2) External activation of Timer 2
  - (3) External trigger of ADC
  - (4) I<sup>2</sup>C-bus
- Port 2: provides the high-order address when expanding the device with external program or data memory.
- Port 3: pins can be configured individually to provide:
  - (1) external interrupt request inputs
  - (2) counter input
  - (3) control signals to read and write to external memories
  - (4) UART input and output
- Port 4 has no alternate function.

To enable a Port pin alternate function, the Port bit latch in its SFR must contain a logic 1.

Each port consists of a latch (Special Function Registers P0 to P4), an output driver and an input buffer. Ports 1,2,3,4 have internal pull-ups. Figure 4(a) shows that the

strong transistor p1 is turned on for only 2 oscillator periods after a 0-to-1 transition in the port latch. When on, it turns on p3 (a weak pull-up) through the inverter. This inverter and p3 form a latch which hold the 1. In Port 0 the pull-up p1 is only on when emitting 1s for external memory access. Writing a 1 to a Port 0 bit latch leaves both output transistors switched off so the pin can be used as a high-impedance input.

#### 1.2.2 Port Options

Thirty-eight of the 40 parallel port pins (excluding P1.6 and P1.7, fixed option 2s) may be individually configured with one of the following options (see Figure 4):

##### Option 1: Standard Port;

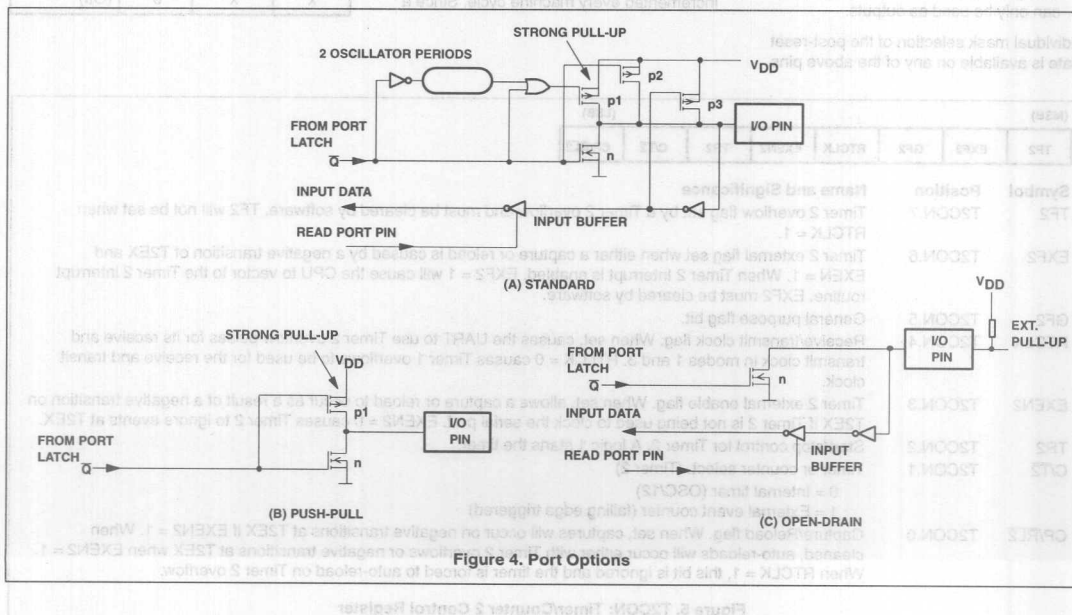
quasi-bidirectional I/O with pull-up. The strong booster pull-up p1 is turned on for two oscillator periods after a 0-to-1 transition in the port latch (see Figure 4(a)).

##### Option 2: Open drain;

quasi-bidirectional I/O with n-channel open drain output. Use as an output requires the connection of an external pull-up resistor (see Figure 4(c)).

##### Option 3: Push-Pull;

output with drive capability in both polarities. Under this option, pins can only be used as outputs (See Figure 4(b)).





## Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

The definition of port options for port 0 is slightly different. Two cases have to be examined. First, accesses to external memory (EA=0 or access above the built-in memory boundary), second, I/O accesses.

### EXTERNAL MEMORY ACCESSES

#### Option 1:

True 0 and 1 are written as address to the external memory (strong pull-up is used).

#### Option 2:

An external pull-up resistor is needed for external accesses.

#### Option 3:

Not allowed for external memory accesses as the port can only be used as output.

### I/O ACCESSES

#### Option 1:

When writing a 1 to the port-latch, the strong pull-up p1 will be on for 2 oscillator periods. No weak pull-up exists. Without an external pull-up, this option can be used as a high-impedance input.

#### Option 2:

Open drain; quasi-bidirectional I/O with n-channel open drain output. Use as an output requires the connection of an external pull-up resistor (see Figure 4(c)).

#### Option 3:

Push-Pull; output with drive capability in both polarities. Under this option, pins can only be used as outputs.

Individual mask selection of the post-reset state is available on any of the above pins.

Make your selection by appending "S" or "R" to option 1, 2, or 3 above (e.g., 1S for a standard I/O to be set after RESET or 2R for an open-drain I/O to be reset after RESET).

#### Option S:

SET; after reset this pin will be initialized HIGH.

#### Option R:

RESET; after reset this pin will be initialized LOW.

### 1.3 Timer/Event Counter

The 83CL580 contains three 16-bit Timer/Counter registers; Timer 0, Timer 1, and Timer 2 which can perform the following functions:

- Measure time intervals and pulse durations
- Count events
- Generate interrupts requests

Timer 0 and Timer 1 can be independently programmed to operate as follows:

- Mode 0; 8-bit timer or counter with divide-by-32 prescaler
- Mode 1; 16-bit time-interval or event counter
- Mode 2; 8-bit time interval or event counter with automatic reload upon overflow
- Mode 3; Timer 0 establishes TL0 and TH0 as two separate counters.

In the "Timer" function, the register is incremented every machine cycle. Since a

machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In the "Counter" function, the register is incremented in response to a 1-to-0 transition. Since it takes 2 machine cycles (24 oscillator periods) to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. To ensure a given level is sampled, it should be held for at least one full machine cycle.

#### 1.3.1 Timer 2

Timer 2 is a 16-bit Timer/Counter. Like Timers 0 and 1, it can operate either as a timer or as an event counter. This is selected by bit C/T2 in the Special Function Register T2CON (Figure 5). It has three operating modes: "capture", "auto-load" and "baud rate generator", which are selected by bits in T2CON as shown in Table 1.

**Table 1. Timer 2 Operating Modes**

RTCLK	CP/RL2	TR2	MODE
0	0	1	16-Bit Auto-reload
0	1	1	16-Bit Capture
1	x	1	Baud Rate Generator
x	x	0	(Off)

(MSB)				(LSB)			
TF2	EXF2	GF2	RTCLK	EXEN2	TR2	C/T2	CP/RL2
<b>Symbol</b>	<b>Position</b>	<b>Name and Significance</b>					
TF2	T2CON.7	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when RTCLK = 1.					
EXF2	T2CON.6	Timer 2 external flag set when either a capture or reload is caused by a negative transition of T2EX and EXEN = 1. When Timer 2 Interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software.					
GF2	T2CON.5	General purpose flag bit.					
RTCLK	T2CON.4	Receive/transmit clock flag. When set, causes the UART to use Timer 2 overflow pulses for its receive and transmit clock in modes 1 and 3. RTCLK = 0 causes Timer 1 overflows to be used for the receive and transmit clock.					
EXEN2	T2CON.3	Timer 2 external enable flag. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.					
TR2	T2CON.2	Start/stop control for Timer 2. A logic 1 starts the timer.					
C/T2	T2CON.1	Timer or counter select. (Timer 2) 0 = Internal timer (OSC/12) 1 = External event counter (falling edge triggered)					
CP/RL2	T2CON.0	Capture/Reload flag. When set, captures will occur on negative transitions at T2EX if EXEN2 = 1. When cleaned, auto-reloads will occur either with Timer 2 overflows or negative transitions at T2EX when EXEN2 = 1. When RTCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.					

**Figure 5. T2CON: Timer/Counter 2 Control Register**

# Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

In the Capture Mode there are two options which are selected by bit EXEN2 in T2CON. If EXEN2=0, then Timer 2 is a 16-bit timer or counter which upon overflowing sets bit TF2, the Timer 2 overflow bit, which can be used to generate an interrupt. If EXEN2=1, then Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input T2EX causes the current value in the Timer 2 registers, TL2 and TH2, to be captured into registers RCAP2L and RCAP2H, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set, and EXF2, like TF2, can generate an interrupt. The Capture Mode is illustrated in Figure 6.

In the auto-reload mode there are again two options, which are selected by bit EXEN2 in T2CON. If EXEN2=0, then when Timer 2 rolls over it not only sets TF2 but also causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2L and RCAP2H, which are preset by software. If EXEN2=1, the Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input T2EX will also trigger the 16-bit reload and set EXF2.

The auto-reload mode is illustrated in Figure 7.

The baud rate generator mode is selected by RTCLK=1. It will be described in conjunction with the serial port.

## 1.3.2 The Watchdog Timer

In addition to Timer T2 and the standard timers, a watchdog timer consisting of an 11-bit prescaler and an 8-bit timer are also incorporated. See Figure 8.

The timer frequency is derived from the oscillator frequency using the following formula:

$$f_{\text{timer}} = \frac{f_{\text{OSC}}}{(12 \times 2048)}$$

When a timer overflow occurs, the microcontroller is reset and a reset output pulse is generated at pin RST. To prevent a system reset the timer must be reloaded in time by the application software. If the processor suffers a hardware/software malfunction, the software will fail to reload the timer. This failure will produce a reset upon overflow thus preventing the processor running out of control.

The watchdog timer can only be reloaded if the condition flag WLE = PCON.4 has been previously set by software.

At the moment the counter is loaded the condition flag is automatically cleared.

The time interval between the timer's reloading and the occurrence of a reset is dependent upon the reloaded value. For example, this may range from 2ms to 0.5s when using an oscillator frequency of 12MHz.

In the Idle state the watchdog timer and reset circuitry remain active.

The watchdog timer is controlled by the watchdog enable pin (EWN). A logic 0 enables the watchdog timer and disables the Power-down mode. A logic 1 disables and resets the watchdog timer and enables the Power-down mode.

## 1.3.3 Pulse Width Modulated Output

One pulse width modulated output channel is provided in the 83CL580 (Figure 9). This channel outputs pulses of programmable length and interval. The repetition frequency is defined by an 8-bit prescaler PWMP which generates the clock for the counter. The 8-bit counter counts modulo 255, i.e., from 0 to 254 inclusive. The value of the 8-bit counter is compared to the content of the register PWM0. Provided the contents of this registers is greater than the counter value, the output of PWM0 is set LOW. If the content of this register is equal to, or less than the counter value, the output will be HIGH. The pulse-width-ratio is therefore defined by the contents of the register PWM0. The pulse-width-ratio is in the range of 0 to 255/255 and may be programmed in increments of 1/255.

The repetition frequency  $f_{\text{PWM}}$ , at PWM0 output is given by:

$$f_{\text{PWM}} = \frac{f_{\text{OSC}}}{[2 \times (1 + \text{PWMP}) \times 255]}$$

When using an oscillator frequency of 12MHz for example, the above formula gives a repetition frequency range of 92Hz to 23.5kHz.

By loading the PWM0 register with either 00H or FFH, the PWM0 output can be retained at a constant HIGH or LOW level respectively. When loading FFH to the PWM0 register, the 8-bit counter will never actually reach this value.

The PWM0 output pin is driven by push-pull drivers, and is not shared with any other function.

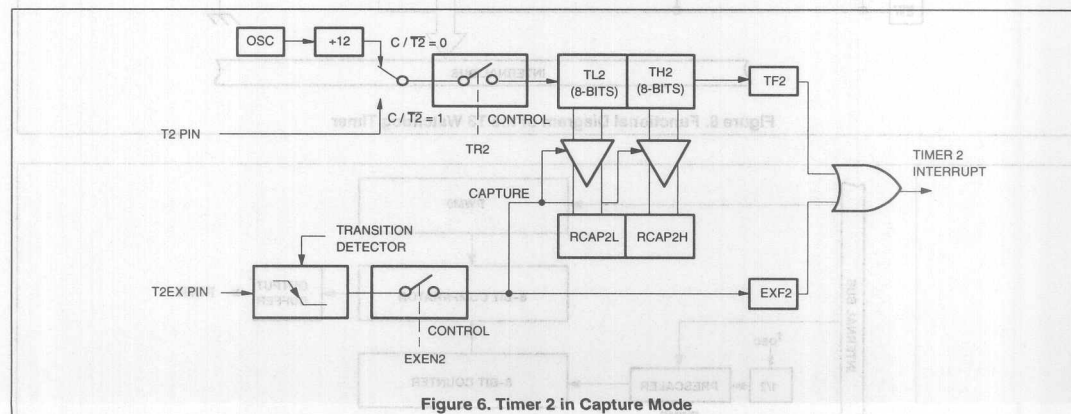
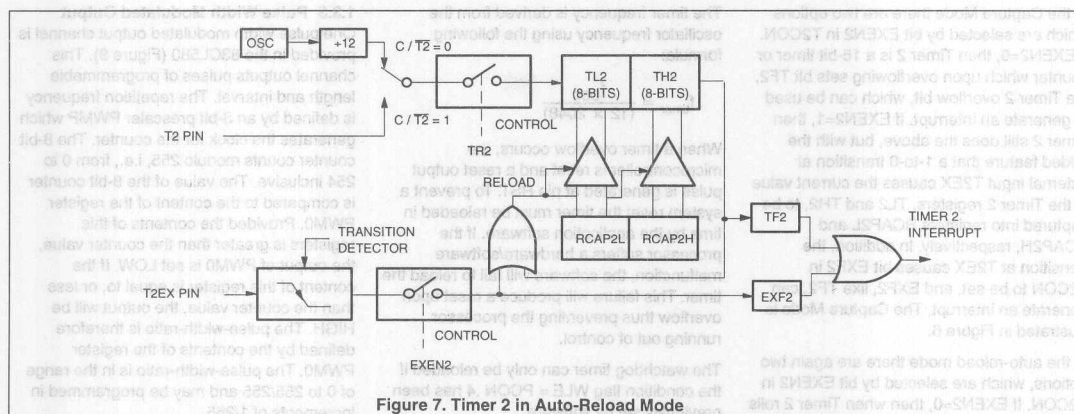


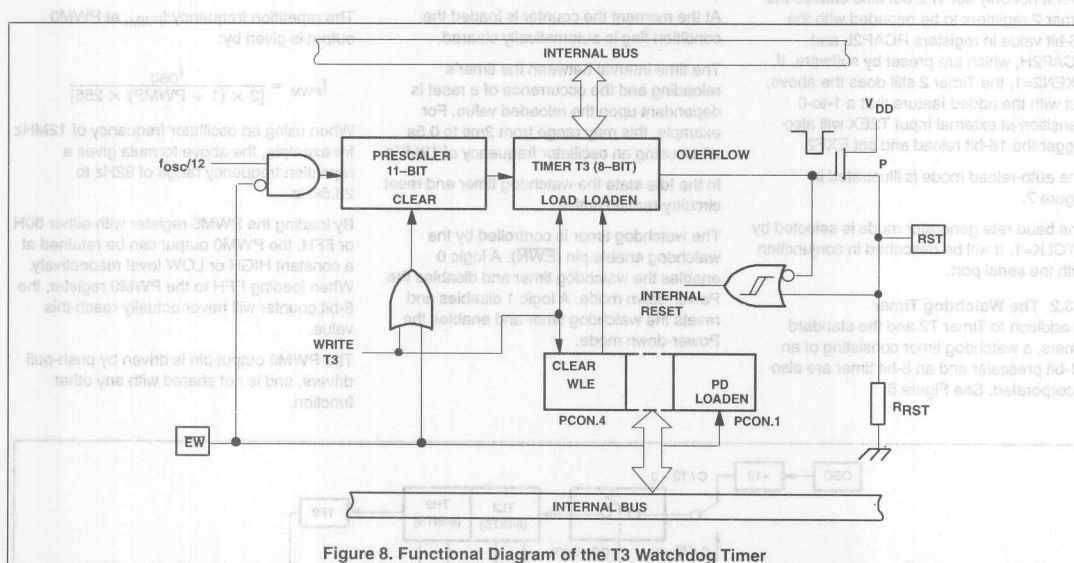
Figure 6. Timer 2 in Capture Mode

## Low-voltage single-chip 8-bit microcontroller

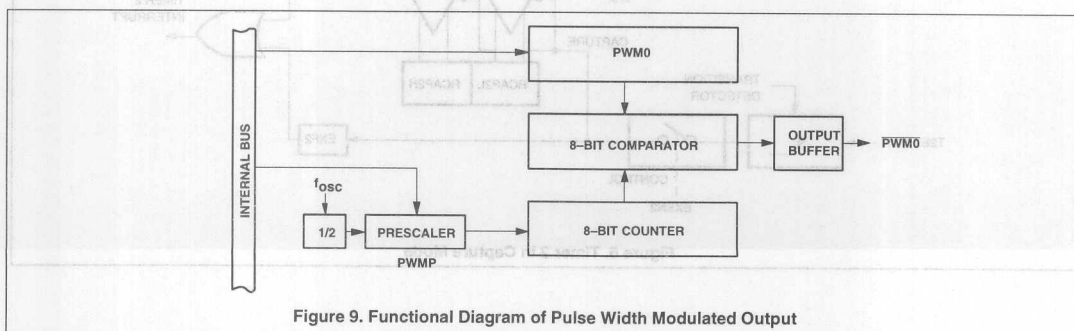
## 80CL580/83CL580



**Figure 7. Timer 2 in Auto-Reload Mode**



**Figure 8. Functional Diagram of the T3 Watchdog Timer**



**Figure 9. Functional Diagram of Pulse Width Modulated Output**

## Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

Prescaler Frequency Control Register  
PWMP

MSB	7	6	5	4	3	2	1	0	LSB
PWMP (FEH)	7	6	5	4	3	2	1	0	

## BIT FUNCTION

PWMP.0-7 Prescaler division factor = (PWMP)+1

## Pulse Width Register PWM0

MSB	7	6	5	4	3	2	1	0	LSB
PWM0 (FCH)	7	6	5	4	3	2	1	0	

## BIT FUNCTION

PWM0 LOW/HIGH ratio of PWM0 signal = (PWM0) / (255 - (PWM0))

## 1.3.4 A/D Converter

The analogue input circuitry consists of an 4-input analogue multiplexer and an ADC with 8-bit resolution. The analogue reference voltage and analogue ground are connected via separate input pins. The conversion is selectable from 24 machine cycles (24µs at 12MHz oscillator frequency) to 48 machine cycles.

The ADC is controlled using the ADCON control register. Input channels are selected by the analogue multiplexer, care of ADCON register bits 0-1.

## ADC Control Register ADCON (C4H)

7	6	5	4	3	2	1	0
PD	ADEX	ADCI	ADCS	CKDIV	AADR1	AADR0	

MSB LSB

## BIT SYMBOL FUNCTION

ADCON.6 ADPD This power down bit switches off the resistor reference to save power even while the CPU is operating.

ADCON.5 ADEX Enable external start of conversion by pin. 0 = Conversion cannot be started externally by STADC. 1 = Conversion can be started externally by STDAC.

ADCON.4 ADCI ADC interrupt flag: this flag is set when an ADC conversion result is ready to be read. An interrupt is invoked if this is enabled. The flag must be cleared by software. It cannot be set by software.

ADCON.3 ADCS ADC start and status: setting this bit starts an ADC conversion. It may be set by software or by the external signal STADC. The ADC logic ensures that this signal is HIGH while the ADC is busy. On completion of the conversion, ADCS is reset after that the interrupt flag ADCI is set. ADCS cannot be reset by software.

ADCI	ADCS	OPERATION
0	0	ADC not busy, a conversion can be started
0	1	ADC busy, start of a new conversion is blocked
1	0	Conversion completed, start of a new conversion is blocked
1	1	Intermediate status for a max of one machine cycle before conversion is completed (ADCI=1, ADCS=0).

## BIT SYMBOL FUNCTION

ADCON.2 CKDIV In order to run the CPU at the maximum frequency (12MHz) and keep the AD timing at low frequency, the setting of this bit divides the CPU clock by a factor 2. 0 = 24 instruction cycles conversion time. 1 = 48 instruction cycles conversion time.

The conversion time includes a sampling time of 6 cycles.

ADCON.1 AADR1 Analogue input select: this binary coded address selects one of the four analogue port bits.

ADCON.0 AADR0 It can only be changed when ADCI and ADCS are both LOW. AADR1 is the most significant bit (11 selects the AD3 analogue input channel).

ADCON.1	ADCON.0	CHANNEL
0	0	AD0
0	1	AD1
1	0	AD2
1	1	AD3

# Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

The completion of the 8-bit ADC conversion is flagged by ADCI in the ADCON register and the result is stored in special function register ADCH (CSH).

An ADC conversion in progress is unaffected by an external or software ADC start. The result of a completed conversion remains unaffected provided ADCI = logic 1. While ADCS = logic 1 or ADCI = logic 1, a new ADC START will be blocked and

consequently lost. An ADC conversion already in progress is aborted when the Power-down mode is entered. The result of a completed conversion (ADCI = logic 1) remains unaffected when entering the Idle and Power-down mode.

The A/D conversion can be started in 3 ways:

- Start in operating mode, continue operating mode.

- Start in operating mode, by setting the ADCS bit, go to idle mode.

- Set the ADEX bit, go to idle mode and start conversion externally via the STADC pin.

For all 3 cases a - c the interrupt flag ADCI is set upon completion of the conversion.

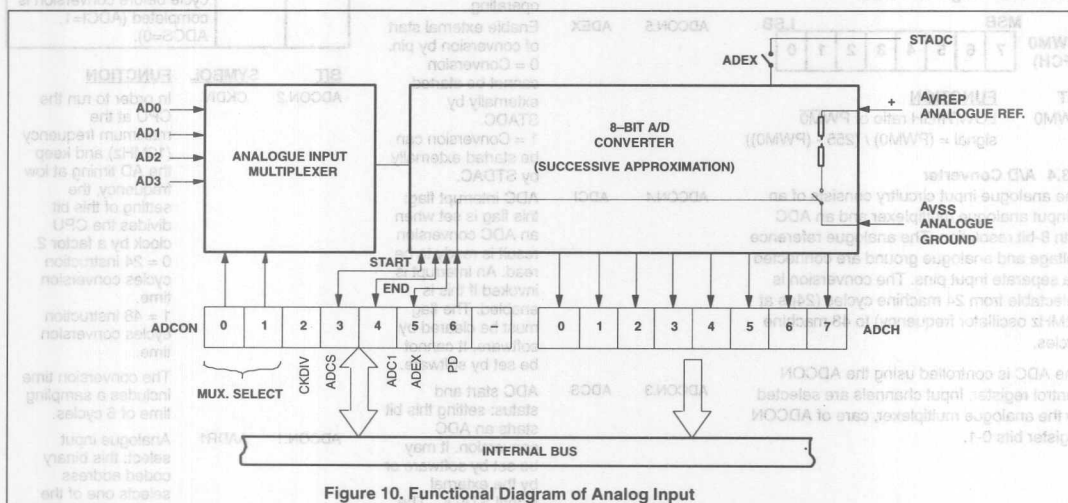


Figure 10. Functional Diagram of Analog Input

CHANNEL	ADCON.0	ADCON.1
AD0	0	0
AD1	1	0
AD2	0	1
AD3	1	1



## 1.4 Idle and Power-down Operation

Idle mode operation permits the interrupt, serial ports, timer blocks and ADC to continue functioning while the clock to the CPU is halted. The following functions remain active during Idle mode. These functions may generate an interrupt or reset and thus end the Idle mode.

- Timer 0, Timer 1, Timer 2, Timer 3
- UART, I<sup>2</sup>C
- External interrupt
- PWM0 (reset, output=HIGH)
- ADC

The Power-down operation freezes the oscillator. The Power-down mode can only be activated by setting the PD bit in the PCON register. The PD bit can only be set if the EWN input is HIGH.

### 1.4.1 Power Control Register (PCON)

These special modes are activated by software via the Special Function Register PCON. Its hardware address is 87H. [PCON is not bit addressable.]

	7	6	5	4	3	2	1	0
SMOD	-	-	WLE	GF1	GF0V	PD	IDL	
MSB				LSB				
<u>BIT</u>	<u>SYMBOL</u>		<u>FUNCTION</u>					
PCON.7	SMOD		Double Baud rate bit. When set to logic 1 baud rate is doubled when the UART serial port is being used in modes 1, 2, or 3.					
PCON.6	-		(reserved)					
PCON.5	-		(reserved)					
PCON.4	WLE		Watchdog load enable. This flag must be set by software prior to loading T3 (watchdog timer). It is cleared when T3 is loaded.					
PCON.3	GF1		General-purpose flag bit					
PCON.2	GF0		General-purpose flag bit					
PCON.1	PD		Power-down bit. Setting this bit activates the Power-down mode. It can only be set if input EWN is high.					
PCON.0	IDL		Idle mode bit. Setting this bit activates the Idle mode.					

If logic 1s are written to PD and IDL at the same time, PD takes precedence. The reset value of PCON is (0X00000).

### 1.4.2 Power-down Mode

The instruction setting PCON.1 is the last executed prior to going into the Power-down mode. In Power-down mode the oscillator is stopped. The contents of the on-chip RAM and SFRs are preserved. The port pins output the values held by their respective SFRs. ALE and PSEN are held LOW.

### 1.4.3 Wake-up Mode

Setting the PD flag in the PCON register forces the controller into the Power-down mode. Setting this flag enables the controller to be woken-up from the Power-down mode with either the external interrupts INT2 / INT8, or a reset operation.

The wake-up operation after power-down in this controller has two basic approaches:

#### 1.4.3.1 Wake-up Using INT2 to INT8

If any of INT2 to INT8 are enabled, the 83CL580 can be woken-up from power-down mode with the external interrupts. To ensure that the oscillator is stable before the controller restarts, the internal clock will remain inactive for 1536 oscillator periods. This is controlled by an on-chip delay counter.

#### 1.4.3.2 Wake-up Using RESET

To wake-up the 83CL580 the RESET pin has to be kept HIGH for a minimum of 24 periods. The on-chip delay counter is inactive. The user has to ensure that the oscillator is stable before any operation is attempted. Figure 11 illustrates the two possibilities for wake-up.

POWER	PORT 4	PORT 3	PORT 2	PORT 1	MODE
Active	Port data	Port data	Port data	Port data	Idle
Active	Port data	Port data	Port data	Port data	Power-down
HIGH	Port data	Port data	Port data	Port data	Wake-up
HIGH	Port data	Port data	Port data	Port data	Wake-up

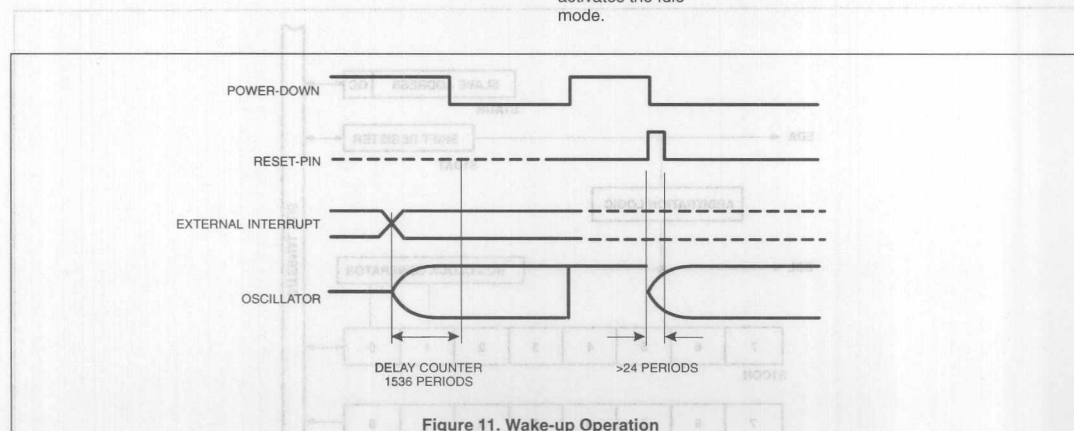


Figure 11. Wake-up Operation

## Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

### 1.4.4 Idle Mode

The instruction that sets PCON.0 is the last instruction executed in the normal operating mode before Idle mode is activated. Once in the Idle mode, the CPU status is preserved in its entirety: the Stack Pointer, Program Counter, Program Status Word, Accumulator, RAM and all other registers maintain their data during Idle mode. The status of the external pins during Idle mode is shown in Table 2.

There are two methods used to terminate the Idle mode. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, terminating Idle mode. The interrupt is serviced, and following the return-from-interrupt instruction RETI, the next instruction to be executed will be the one which follows the instruction that wrote a logic 1 to PCON.0.

Flag bits GF0 and GF1 may be used to determine whether the interrupt was received during normal execution or during Idle mode. For example, the instruction that writes to PCON.0 can also set or clear one or both flag bits. When Idle mode is terminated by an

interrupt, the service routine can examine the status of the flag bits.

The second method of terminating the Idle mode is with an external hardware reset, or an internal reset caused by an overflow of Timer T3. Since the oscillator is still running, the hardware reset is required to be active for two machine cycles (24 oscillator periods) to complete the reset operation.

Reset redefines all SFRs, but does not affect the on-chip RAM.

In the Power-down mode,  $V_{DD}$  may be reduced to minimize power consumption. However, the supply voltage must not be reduced until Power-down mode is active, and must be held active until the oscillator has restarted and stabilized.

The status of the external pins during Idle and Power-down mode is shown in Table 2. If the Power-down mode is activated whilst accessing external memory, port data held in the Special Function Register P2 is restored to Port 2. If the data is a logic 1, the port pin is held HIGH during the Power-down mode by the strong pull-up transistor p1 (see Figure 4(a)).

### 1.5 I<sup>2</sup>C Bus Serial I/O

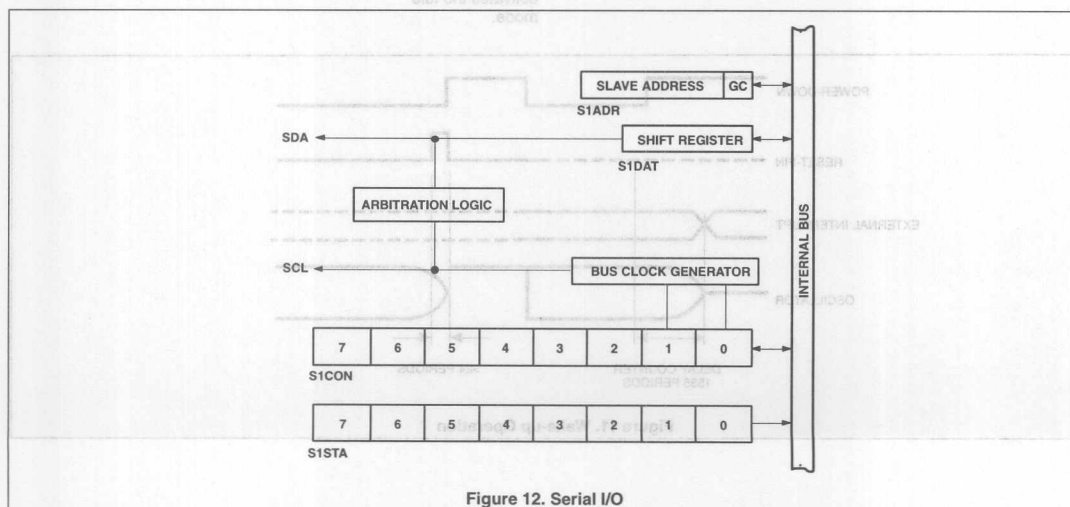
The serial port supports the twin line I<sup>2</sup>C-bus. The I<sup>2</sup>C-bus consists of a data line (SDA) and a clock line (SCL). These lines also function as I/O port lines P1.7 and P1.6 respectively. The system is unique because data transport, clock generation, address recognition and bus control arbitration are all controlled by hardware. The I<sup>2</sup>C-bus serial I/O has complete autonomy in byte handling and operates in four modes:

- Master transmitter
- Master receiver
- Slave transmitter
- Slave receiver

These functions are controlled by the S1CON register. S1STA is the status register whose contents may also be used as a vector to various service routines. S1DAT is the data shift register and S1ADR the slave address register. Slave address recognition is performed by hardware.

**Table 2. Status of the External Pins During Idle and Power-down Modes**

MODE	MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3	PORT 4	PWM0
Idle	Internal	1	1	Port data	Port data	Port data	Port data	Port data	Active
Idle	External	1	1	Floating	Port data	Address	Port data	Port data	Active
Power-down	Internal	0	0	Port data	Port data	Port data	Port data	Port data	HIGH
Power-down	External	0	0	Floating	Port data	Port data	Port data	Port data	HIGH



**Figure 12. Serial I/O**

# Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

## Serial Control Register S1CON (D8H)

CR2	ENS1	STA	STO	SI	AA	CR1	CR0
-----	------	-----	-----	----	----	-----	-----

**CR0, CR1, CR2** These three bits determine the serial clock frequency when SIO is in a master mode. See Table 3.

**AA** Assert acknowledge bit. When the AA flag is set, an acknowledge (low level to SDA) will be returned during the acknowledge clock pulse on the SCL line when:

- own slave address is received
- general call address is received (S1ADR.0=1)
- data byte received while device is programmed as master
- data byte received while device is a selected slave

With AA = 0, no acknowledge will be returned. Consequently, no interrupt is requested when the "own slave address" or general call address is received.

**SI** SIO interrupt flag. When the S1 flag is set, an acknowledge is returned after any one of the following conditions:

- a start condition is generated in master mode
- own slave address received during AA = 1
- general call address received while S1ADR.0 and AA = 1
- data byte received or transmitted in master mode (even if arbitration is lost)
- data byte received or transmitted as selected slave
- stop or start condition received as selected slave receiver or transmitter

**STO** STOP flag. With this bit set while in master mode a STOP condition is generated. When a STOP condition is detected on the bus, the SIO hardware clears the STO flag. In the slave mode, the STO flag may also be set to recover from an error condition. In this case no STOP condition is transmitted to the I<sup>2</sup>C-bus. However, the SIO hardware behaves as if a STOP condition has been received and releases SDA and SCL. The SIO then switches to the "not addressed" slave receiver mode. The STO flag is automatically cleared by hardware.

**STA** START flag. When the STA bit is set in slave mode, the SIO hardware checks the status of the I<sup>2</sup>C-bus and generates a START condition if the bus is free. If STA is set while the SIO is in master mode, SIO transmits a repeated START condition.

**ENS1** When ENS1 = 0, the SIO is disabled. The SDA and SCL outputs are in a high-impedance state; P1.6 and P1.7 function as open drain ports. When ENS = 1, the SIO is enabled. The P1.6 and P1.7 port latches must be set to logic 1.

**Table 3. SCL Frequency**

CR2	CR1	CR0	f <sub>osc</sub> Divided By	Bit Rate (kHz) at f <sub>osc</sub>		
				3.58MHz	6MHz	12MHz
0	0	0	256	14.0	23.4	46.9
0	0	1	224	16.0	26.8	53.6
0	1	0	192	18.6	31.3	62.5
0	1	1	160	22.4	37.5	75.0
1	0	0	960	3.73	6.25	12.5
1	0	1	120	29.8	50	100
1	1	0	60	59.7	100	—
1	1	1	not allowed	—	—	—

# Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

## Status Register S1STA (D9H)

SC4	SC3	SC2	SC1	SC0	0	0	0
-----	-----	-----	-----	-----	---	---	---

S1STA is an 8-bit read-only special function register. S1STA.3-S1STA.7 hold a status code. S1STA.0-S1STA.2 are held LOW. The contents of S1STA may be used as a vector to a service routine. This optimizes response time of the software and consequently that of the I<sup>2</sup>C-bus.

The following is a list of the status code:

### Abbreviations used:

SLA:	7-bit slave address
R:	Read bit
W:	Write bit
ACK:	Acknowledgement (acknowledge bit = 0)
ACKNOT:	Not Acknowledgement (acknowledge bit = 1)
DATA:	8-bit byte to or from the I <sup>2</sup> C bus
MST:	Master
SLV:	Slave
TRX:	Transmitter
REC:	Receiver

### MST/TRX Mode

S1STA Value

08H	A START condition has been transmitted
10H	A repeated START condition has been transmitted
18H	SLA and W have been transmitted, ACK received
20H	SLA and W have been transmitted, ACKNOT received
28H	DATA of S1DAT has been transmitted, ACK received
30H	DATA of S1DAT has been transmitted, ACKNOT received
38H	Arbitration lost in SLA, R/W or DATA

### MST/REC Mode

S1STA Value

08H	A START condition has been transmitted
10H	A repeated START condition has been transmitted
38H	Arbitration lost while returning ACKNOT
40H	SLA and R have been transmitted, ACK received
48H	SLA and R have been transmitted, ACKNOT received
50H	DATA has been received, ACK returned
58H	DATA has been received, ACKNOT returned

## SLV/REC Mode

S1STA Value

60H	Own SLA and W have been received, ACK returned
68H	Arbitration lost in SLA, RW as MST. Own SLA and W have been received, ACK returned.
70H	General Call has been received, ACK returned
78H	Arbitration lost in SLA, RW as MST. General Call has been received.
80H	Previously addressed with own SLA. DATA byte received, ACK returned.
88H	Previously addressed with own SLA. DATA byte has been received, ACKNOT has been returned.
90H	Previously addressed with General Call. DATA byte has been received, ACK has been returned.
98H	Previously addressed with General Call. DATA byte has been received, ACKNOT has been returned.
A0H	A STOP condition or repeated START condition has been received while still addressed as SLV/REC or SLV/TRX.

## SLV/TRX Mode

S1STA Value

A8H	Own SLA and R have been received, ACK returned.
B0H	Arbitration lost in SLA, R/W as MST. Own SLA and R have been received, ACK returned.
B8H	DATA byte has been transmitted, ACK received.
C0H	DATA byte has been transmitted, ACKNOT received.
C8H	Last data byte has been transmitted (AA = logic 0), ACK received.

## Miscellaneous

S1STA Value

00H	Bus error during MST mode or selected SLV mode, due to an erroneous START or STOP condition.
F8H	No relevant state information available, SI = 0.

## Data Shift Register S1DAT (DAH)

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

This register contains the serial data to be transmitted or data that has just been received. Bit 7 is transmitted or received first; i.e., data is shifted from left to right.

## Own Address Register S1ADR (DBH)

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

This 8-bit register may be loaded with the 7-bit address to which the controller will respond when programmed as a slave receiver/transmitter. The LSB bit (GC) is used to determine whether the general CALL address is recognized.

S1ADR.0, GC: 0 = general CALL address is not recognized.  
1 = general CALL is recognized.  
S1ADR.7 - 1: own slave address

## 1.6 Standard serial interface SIO0: UART

This serial port is full duplex, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the register. (However, if the first byte still has not been read by the time reception of the second byte is complete, one of the bytes will be lost). The serial port receive and transmit registers are both accessed at Special Function Register S0BUF. Writing to S0BUF loads the transmit register, and reading S0BUF accesses a physically separate receive register.

The serial port can operate in 4 modes:

**Mode 0:** Serial data enters and exits through RxD. TxD outputs the shift clock. 8 bits are transmitted/received (LSB first). The baud is fixed at 1/12 the oscillator frequency.

**Mode 1:** 10 bits are transmitted (through TxD) or received (through RxD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in Special Function Register SCON. The baud rate is variable.

## Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

**Mode 2:** 11 bits are transmitted (through TxD) or received (through RxD): start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On Transmit, the 9th data bit (TB8 in SCON) can be assigned the value of 0 or 1. Or, for example, the parity bit (P, in the PSW) could be moved into TB8. On receive, the 9th data bit goes into RB8 in Special Function Register SCON, while the stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 the oscillator frequency.

**Mode 3:** 11 bits are transmitted (through TxD) or received (through RxD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit and a stop bit (1). In fact, Mode 3 is the same as Mode 2 in all respects except baud rate. The baud rate in Mode 3 is variable.

In all four modes, transmission is initiated by any instruction that uses S0BUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit if REN = 1.

**1.6.1 Multiprocessor Communications**  
Modes 2 and 3 have a special provision for multiprocessor communications. In these modes, 9 data bits are received. The 9th one goes into RB8. Then comes a stop bit. The port can be programmed such that when the stop bit is received, the serial port interrupt will be activated only if RB8 = 1. This feature is enabled by setting bit SM2 in SCON. A way to use this feature in multiprocessor systems is as follows.

When the master processor wants to transmit a block of data to one of several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte. With SM2 = 1, no slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave will clear its SM2 bit and prepare to receive the data bytes that will be coming. The slaves that were not being addressed leave their SM2s set and go on about their business, ignoring the coming data bytes.

SM2 has no effect in Mode 0, and in Mode 1 can be used to check the validity of the stop bit. In a Mode 1 reception, if SM2 = 1, the

receive interrupt will not be activated unless a valid stop bit is received.

### 1.6.2 Serial port control register

The serial port control and status register is the Special Function Register S0CON, shown in Figure 13. The register contains not only the mode selection bits, but also the 9th data bit for transmit and receive (TB8 and RB8), and the serial port interrupt bits (TI and RI).

**Baud Rates**—The baud rate in Mode 0 is fixed: Mode 0 Baud Rate = Oscillator Frequency / 12. The baud rate in Mode 2 depends on the value of bit SMOD in Special Function Register PCON. If SMOD = 0 (which is the value on reset), the baud rate is 1/64 the oscillator frequency. If SMOD = 1, the baud rate is 1/32 the oscillator frequency.

Mode 2 Baud Rate =  $(2^{\text{SMOD}} / 64) \times (\text{Oscillator Frequency})$

The baud rates in Modes 1 and 3 are determined by the Timer 1 or Timer 2 overflow rate.

### Using Timer 1 to Generate Baud

**Rates**—When Timer 1 is used as the baud rate generator, the baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate and the value of SMOD as follows:

Modes 1,3 Baud Rate =  $(2^{\text{SMOD}} / 32) \times (\text{Timer 1 Overflow Rate})$

The Timer 1 interrupt should be disabled in this application. The Timer itself can be configured for either "timer" or "counter" operation, and in any of its 3 running modes. In most typical applications, it is configured for "timer operation, in the auto-reload mode (high nibble of TMOD = 0010B). In that case the baud rate is given by the formula:

Mode 1, 3 Baud Rate =  $\{(2^{\text{SMOD}} / 32) \times (\text{Oscillator Frequency})\} / \{12 \times (256 - (\text{TH1}))\}$

One can achieve very low baud rates with Timer 1 by leaving the Timer 1 interrupt enabled, and configuring the Timer to run as a 16-bit timer (high nibble of TMOD = 0001B), and using the Timer 1 interrupt to do a 16-bit software reload. Table 4 lists various commonly used baud rates and how they can be obtained from Timer 1.

**Using Timer 2 to Generate Baud Rates**—Timer 2 is selected as the baud rate generator by setting RTCLK in T2CON (Figure 14).

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by software.

Now, the baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate as follows:

Modes 1,3 Baud Rate =  $(\text{Timer 2 Overflow Rate}) / 16$

The Timer 2 can be configured for either "timer" or "counter" operation. In the most typical applications, it is configured for "timer" operation (C/T2 = 0). "Timer" operation is a little different for Timer 2 when it's being used as a baud rate generator. Normally, as a timer is would increment every machine cycle (thus at 1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time (thus at 1/2 the oscillator frequency). In that case the baud rate is given by the formula:

Modes 1,3 Baud Rate =  $(\text{Oscillator Frequency}) / \{32 \times (65536 - (\text{RCAP2H}, \text{RCAP2L}))\}$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

Note that a rollover does not set TF2, and will not generate an interrupt. Therefore, the Timer 2 interrupt does not have to be disabled when Timer 2 is in the baud rate generator mode. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt, if desired.

It should be noted that when Timer 2 is running (TR2 = 1) in "timer" function in the baud rate generator mode, one should not try to read or write TH2 or TL2. Under these conditions the Timer is being incremented every state time, and the results of a read or write may not be accurate. The RCAP registers may be read, but shouldn't be written to, because a write might overlap a reload and cause write and/or reload errors. Turn the Timer off (clear TR2) before accessing the Timer 2 or RCAP register, in this case.



## Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

MSB						LSB	
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Where SM0, SM1 specify the serial port mode, as follows:

SM0	SM1	MODE	DESCRIPTION	BAUD RATE
0	0	0	Shift register	$f_{osc} / 12$
0	1	1	8-bit UART	variable
1	0	2	9-bit UART	$f_{osc} / 64$ or $f_{osc} / 32$
1	1	3	9-bit variable UART	

**SM2** enables the multiprocessor communication feature in Modes 2 and 3. In Mode 2 or 3, if SM2 is set to 1 then R1 will not be activated if the received ninth data bit (RB8) is 0. In Mode 1, if SM2=1 then R1 will not be activated if a valid stopbit was not received. In Mode 0, SM2 should be 0.

**REN** enables serial reception. Set by software to enable reception. Clear by software to disable reception.

**TB8** is the ninth data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired.

**RB8** in Modes 2 and 3, is the ninth data bit that was received. In Mode 1, if SM2=0, RB8 is the stop bit that was received. In Mode 0, RB8 is not used.

**TI** is transmit interrupt flag. Set by hardware at the end of the eighth bit time in Mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.

**RI** is receive interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM2). Must be cleared by software.

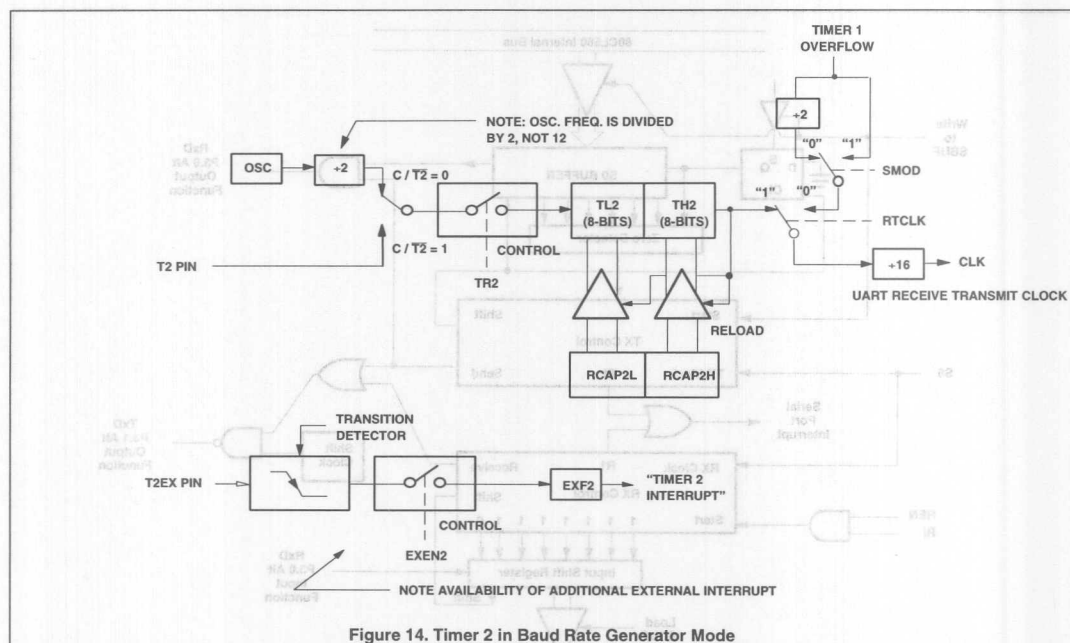
**Figure 13. Serial Port Control (SCON) Register**

Figure 13. Serial Port Control (SCON) Register

Table 4. Timer 1 Generated Commonly Used Baud Rates

BAUD RATE	$f_{osc}$	SMOD	C/T	TIMER 1 MODE	RELOAD VALUE
Mode 0 Max: 1MHz	12MHz	X	X	X	X
Mode 2 Max: 375K	12MHz	1	X	X	X
Mode 1, 3: 62.5K	12MHz	1	0	2	FFH
19.2K	11.059MHz	1	0	2	FDH
9.6K	11.059MHz	0	0	2	FDH
4.8K	11.059MHz	0	0	2	FAH
2.4K	11.059MHz	0	0	2	F4H
1.2K	11.059MHz	0	0	2	E8H
137.5K	11.986MHz	0	0	2	1DH
110K	6MHz	0	0	2	72H
110K	12MHz	0	0	1	FE8H

Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580



**Figure 14. Timer 2 in Baud Rate Generator Mode**

## Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

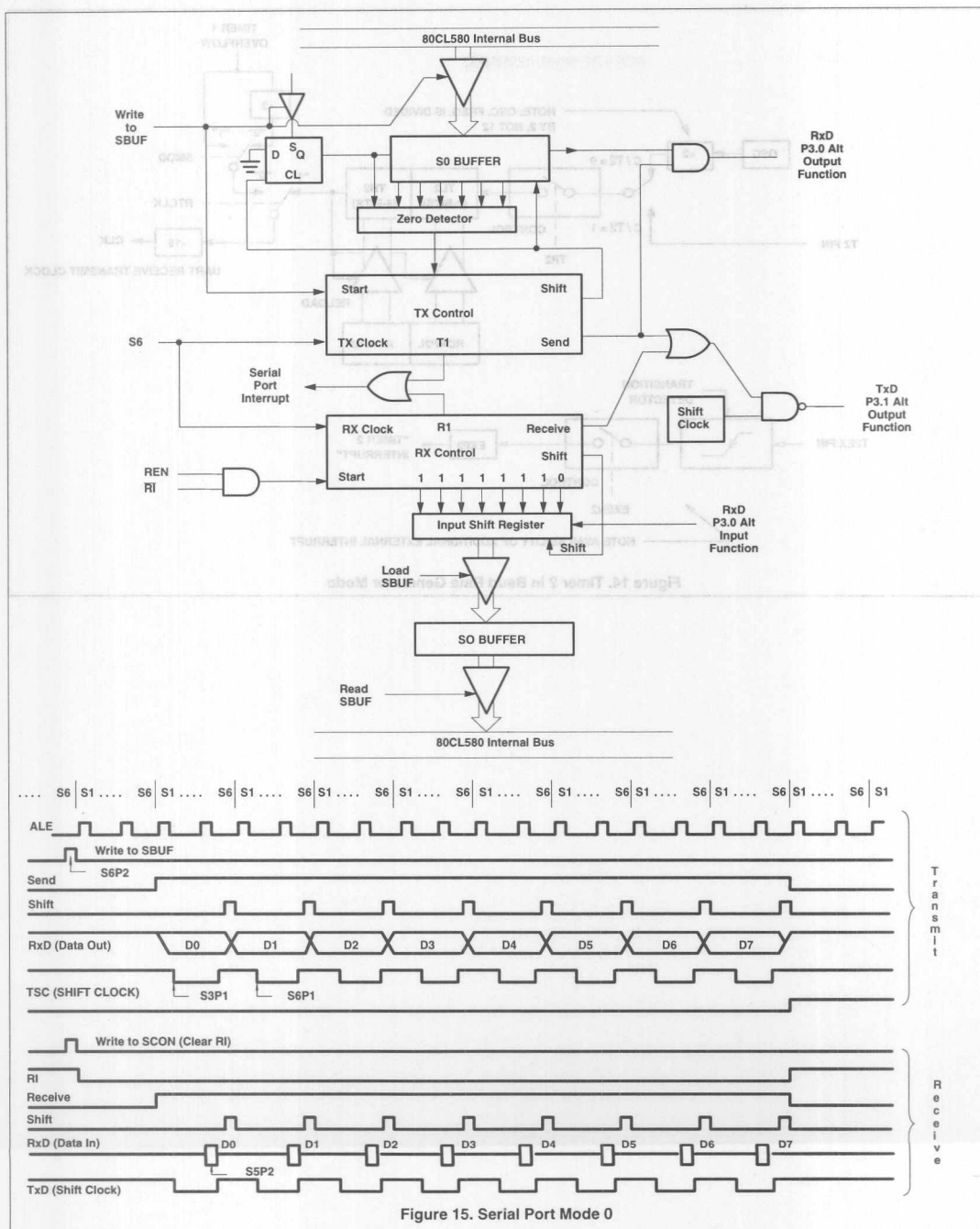


Figure 15. Serial Port Mode 0

## Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

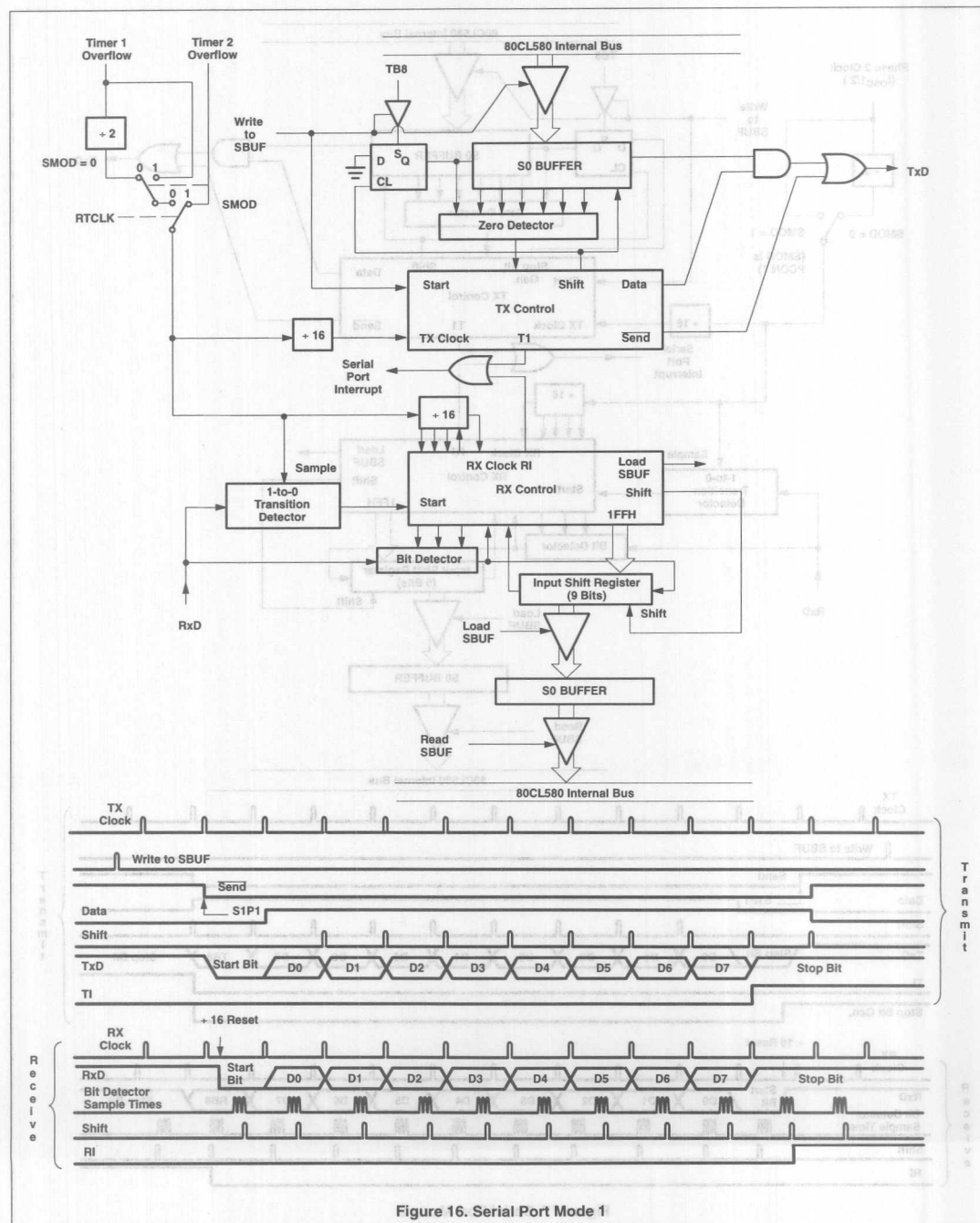


Figure 16. Serial Port Mode 1

## Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

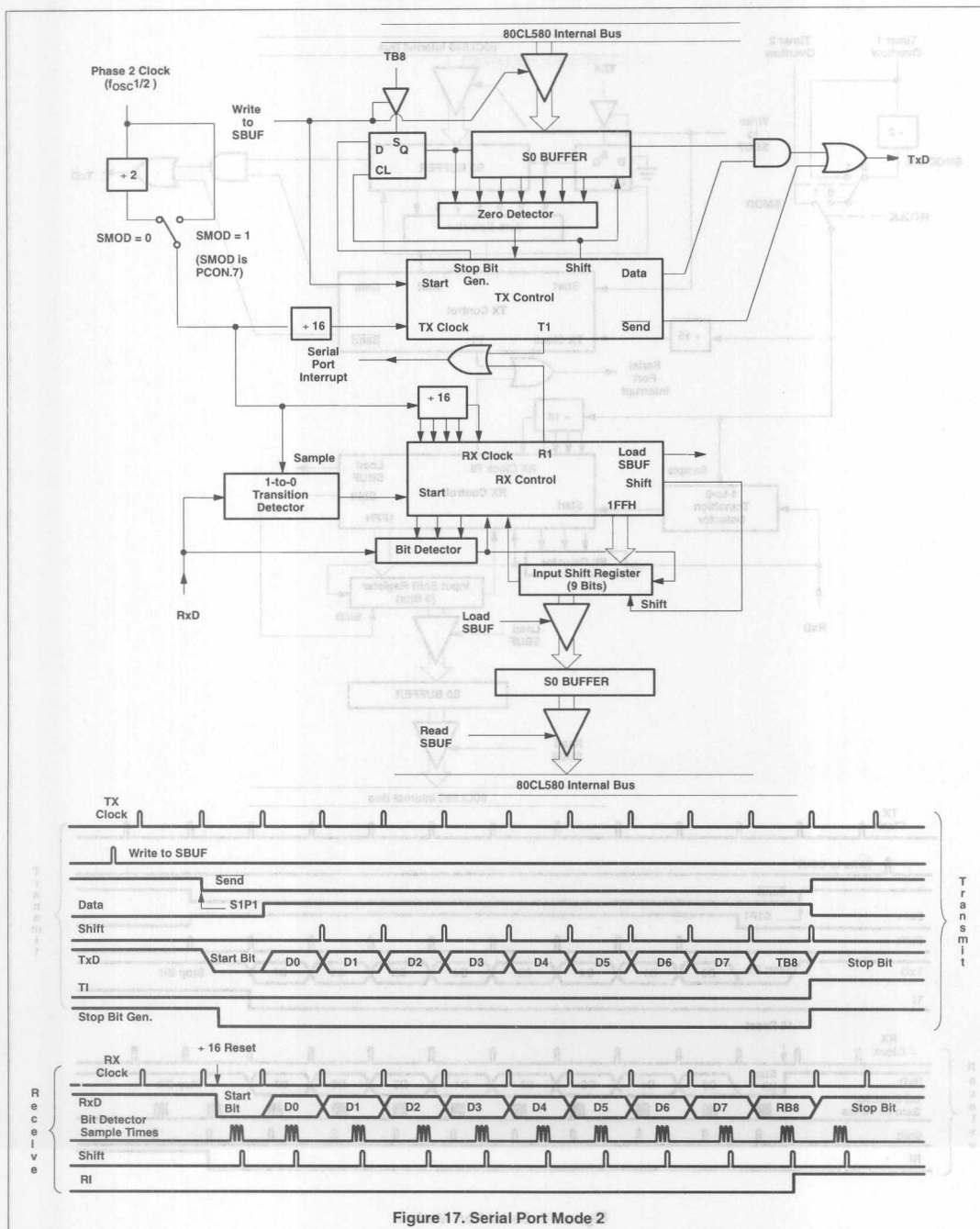


Figure 17. Serial Port Mode 2



## Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

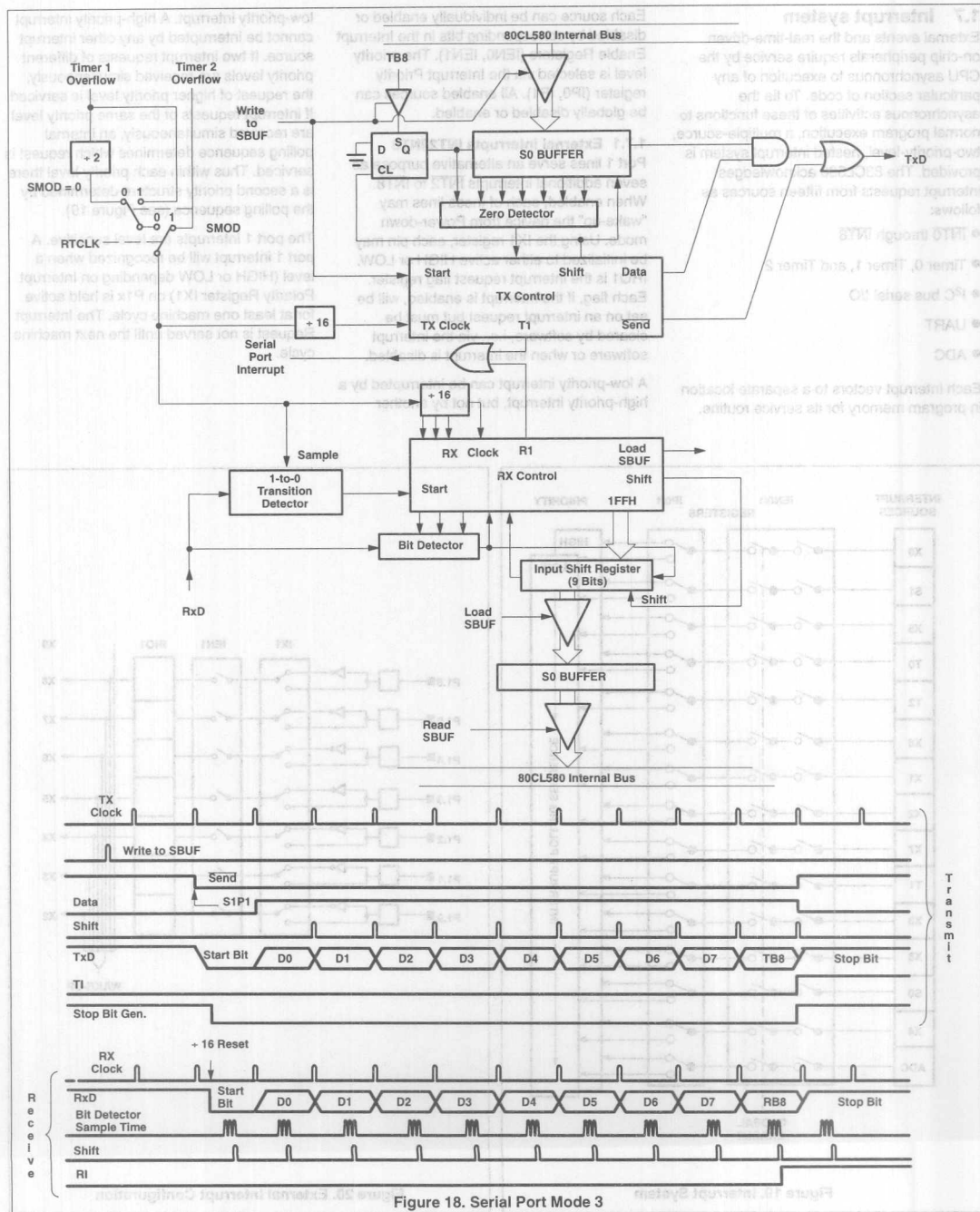


Figure 18. Serial Port Mode 3

# Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

## 1.7 Interrupt system

External events and the real-time-driven on-chip peripherals require service by the CPU asynchronous to execution of any particular section of code. To tie the asynchronous activities of these functions to normal program execution, a multiple-source, two-priority-level, nested interrupt system is provided. The 83CL580 acknowledges interrupt requests from fifteen sources as follows:

- INT0 through INT8
- Timer 0, Timer 1, and Timer 2
- I<sup>2</sup>C bus serial I/O
- UART
- ADC

Each interrupt vectors to a separate location in program memory for its service routine.

Each source can be individually enabled or disabled by corresponding bits in the Interrupt Enable Registers (IEN0, IEN1). The priority level is selected via the Interrupt Priority register (IP0, IP1). All enabled sources can be globally disabled or enabled.

### 1.7.1 External interrupts INT2/INT8

Port 1 lines serve an alternative purpose as seven additional interrupts INT2 to INT8. When enabled, each of these lines may "wake-up" the device from Power-down mode. Using the IX1 register, each pin may be initialized to either active HIGH or LOW. IRQ1 is the interrupt request flag register. Each flag, if the interrupt is enabled, will be set on an interrupt request but must be cleared by software, i.e., via the interrupt software or when the interrupt is disabled.

A low-priority interrupt can be interrupted by a high-priority interrupt, but not by another

low-priority interrupt. A high-priority interrupt cannot be interrupted by any other interrupt source. If two interrupt requests of different priority levels are received simultaneously, the request of higher priority level is serviced. If interrupt requests of the same priority level are received simultaneously, an internal polling sequence determines which request is serviced. Thus within each priority level there is a second priority structure determined by the polling sequence (see Figure 19).

The port 1 interrupts are level sensitive. A port 1 interrupt will be recognized when a level (HIGH or LOW) depending on Interrupt Polarity Register IX1 on P1x is held active for at least one machine cycle. The Interrupt Request is not served until the next machine cycle.

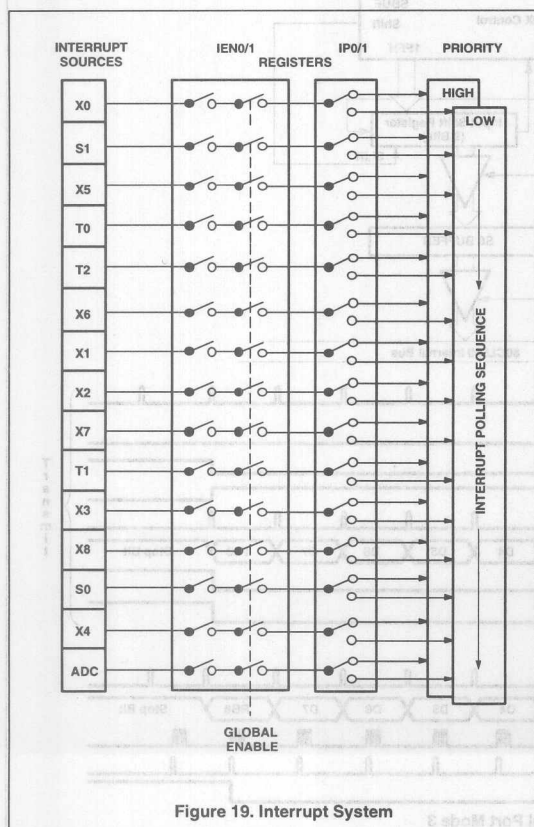


Figure 19. Interrupt System

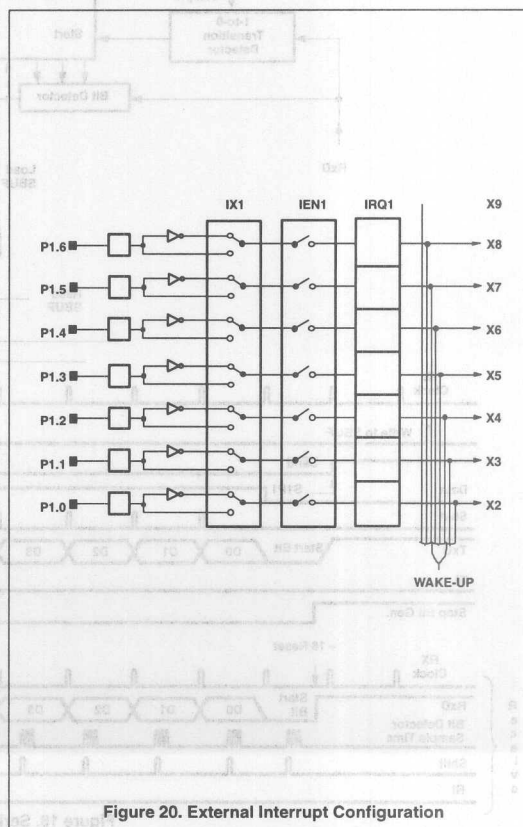


Figure 20. External Interrupt Configuration

## Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

## Interrupt Enable Register

## IEN0, IEN1

IEN0 (A8H)	EA	ET2	ES1	ES0	ET1	EX1	ET0	EX0
------------	----	-----	-----	-----	-----	-----	-----	-----

BIT	SYMBOL	FUNCTION
IEN0.7	EA	General enable/disable control 0 = no interrupt is enabled; 1 = any individually enabled interrupt will be accepted
IEN0.6	ET2	Enable T2 interrupt
IEN0.5	ES1	Enable I <sup>2</sup> C interrupt
IEN0.4	ES0	Enable UART SIO interrupt
IEN0.3	ET1	Enable Timer T1 interrupt
IEN0.2	EX1	Enable external interrupt 1
IEN0.1	ET0	Enable Timer T0 interrupt
IEN0.0	EX0	Enable external interrupt 0

IEN1 (E8H)	EAD	EX8	EX7	EX6	EX5	EX4	EX3	EX2
------------	-----	-----	-----	-----	-----	-----	-----	-----

BIT	SYMBOL	FUNCTION
IEN1.7	EAD	Enable ADC interrupt
IEN1.6	EX8	Enable external interrupt 8
IEN1.5	EX7	Enable external interrupt 7
IEN1.4	EX6	Enable external interrupt 6
IEN1.3	EX5	Enable external interrupt 5
IEN1.2	EX4	Enable external interrupt 4
IEN1.1	EX3	Enable external interrupt 3
IEN1.0	EX2	Enable external interrupt 2

where 0 = interrupt disabled  
1 = interrupt enabled

## Interrupt Priority Register

## IP0, IP1

IP0 (B8H)	-	PT2	PS1	PS0	PT1	PX1	PT0	PX0
-----------	---	-----	-----	-----	-----	-----	-----	-----

BIT	SYMBOL	FUNCTION
IP0.7	-	Unused
IP0.6	PT2	Timer 2 interrupt priority level
IP0.5	PS1	I <sup>2</sup> C interrupt priority level
IP0.4	PS0	UART SIO interrupt priority level
IP0.3	PT1	Timer 1 interrupt priority level
IP0.2	PX1	External interrupt 1 priority level
IP0.1	PT0	Timer 0 interrupt priority level
IP0.0	PX0	External interrupt 0 priority level

IP1 (F8H)	PX9	PX8	PX7	PX6	PX5	PX4	PX3	PX2
-----------	-----	-----	-----	-----	-----	-----	-----	-----

BIT	SYMBOL	FUNCTION
IP1.7	PADC	ADC interrupt priority level
IP1.6	PX8	External interrupt 8 priority level
IP1.5	PX7	External interrupt 7 priority level
IP1.4	PX6	External interrupt 6 priority level
IP1.3	PX5	External interrupt 5 priority level
IP1.2	PX4	External interrupt 4 priority level
IP1.1	PX3	External interrupt 3 priority level
IP1.0	PX2	External interrupt 2 priority level

Interrupt priority is as follows:  
0 = low priority  
1 = high priority

## Interrupt Polarity Register

## IX1

IX1 (E9H)	-	IL8	IL7	IL6	IL5	IL4	IL3	IL2
-----------	---	-----	-----	-----	-----	-----	-----	-----

Writing either a "1" or "0" to an IX1 register bit sets the polarity level of the corresponding external interrupt to active HIGH or LOW respectively.

BIT	SYMBOL	FUNCTION
IX1.7	-	Unused
IX1.6	IL8	External interrupt 8 polarity level
IX1.5	IL7	External interrupt 7 polarity level
IX1.4	IL6	External interrupt 6 polarity level
IX1.3	IL5	External interrupt 5 polarity level
IX1.2	IL4	External interrupt 4 polarity level
IX1.1	IL3	External interrupt 3 polarity level
IX1.0	IL2	External interrupt 2 polarity level

## Interrupt Request Flag Register

## IRQ1

IRQ1 (C0H)	-	IQ8	IQ7	IQ6	IQ5	IQ4	IQ3	IQ2
------------	---	-----	-----	-----	-----	-----	-----	-----

BIT	SYMBOL	FUNCTION
IRQ1.7	-	Unused
IRQ1.6	IQ8	External interrupt 8 request flag
IRQ1.5	IQ7	External interrupt 7 request flag
IRQ1.4	IQ6	External interrupt 6 request flag
IRQ1.3	IQ5	External interrupt 5 request flag
IRQ1.2	IQ4	External interrupt 4 request flag
IRQ1.1	IQ3	External interrupt 3 request flag
IRQ1.0	IQ2	External interrupt 2 request flag

# Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

## 1.7.2 Interrupt vectors

(highest)	Vector	Source
X0	0003H	external 0
S1	002BH	I <sup>2</sup> C port
X5	0053H	external 5
T0	000BH	timer 0
T2	0033H	timer 2
X6	005BH	external 6
X1	0013H	external 1
X2	003BH	external 2
X7	0063H	external 7
T1	001BH	timer 1
X3	0043H	external 3
X8	006BH	external 8
S0	0023H	UART
X4	004BH	external 4
ADC	0073H	ADC
(lowest)		

**Interrupt Priority**—Each interrupt priority source can be set to either high or low priority. If both priorities are requested simultaneously, the controller will branch to the high priority vector.

A low priority interrupt can only be interrupted by a high priority interrupt. A high priority interrupt routine cannot be interrupted.

## 1.7.3 Related Registers

The following registers are used in conjunction with the interrupt system:

REGISTER	FUNCTION	SFR ADDRESS
IX1	Interrupt polarity register	E9H
IRQ1	Interrupt request flag register	C0H
IEN0	Interrupt enable register	A8H
IEN1	Interrupt enable register (INT2-INT8, ADC)	E8H
IP0	Interrupt priority register	B8H
IP1	Interrupt priority register (INT2-INT8, ADC)	F8H

## 1.8 Oscillator Circuitry

The on-chip oscillator circuitry of the 83CL580 is a single-stage inverting amplifier biased by an internal feedback resistor (Figure 21). For operation as a standard quartz oscillator, no external components are needed, except at 32kHz. When using external capacitors, ceramic resonators, coils and RC networks to drive the oscillator, five different configurations are supported (see Figure 22 and oscillator options).

In the Power-down mode the oscillator is stopped and XTAL1 is pulled HIGH. The

oscillator inverter is switched off to ensure no current will flow regardless of the voltage at XTAL1. To drive the device with an external clock source, apply the external clock signal to XTAL1, and leave XTAL2 to float, as shown in Figure 18(f). There are no requirements on the duty cycle of the external clock, since the input to the internal clocking circuitry is buffered by a flip-flop.

The following options are provided for optimum on-chip oscillator performance. Please state option when ordering.

### 1.8.1 Oscillator Options (see Figure 22)

- Osc 1: Figure 22(c). An option for 32kHz clock applications with external trimmer for frequency adjustment. A 4.7M $\Omega$  bias resistor is needed for use in parallel with the crystal.
- Osc 2: Figure 22(e). An option for low-power, low-frequency operations using LC components or quartz.
- Osc 3: An option for medium frequency range applications.
- Osc 4: An option for high frequency range applications.
- RC: Figure 22(g). An option for an RC oscillator.

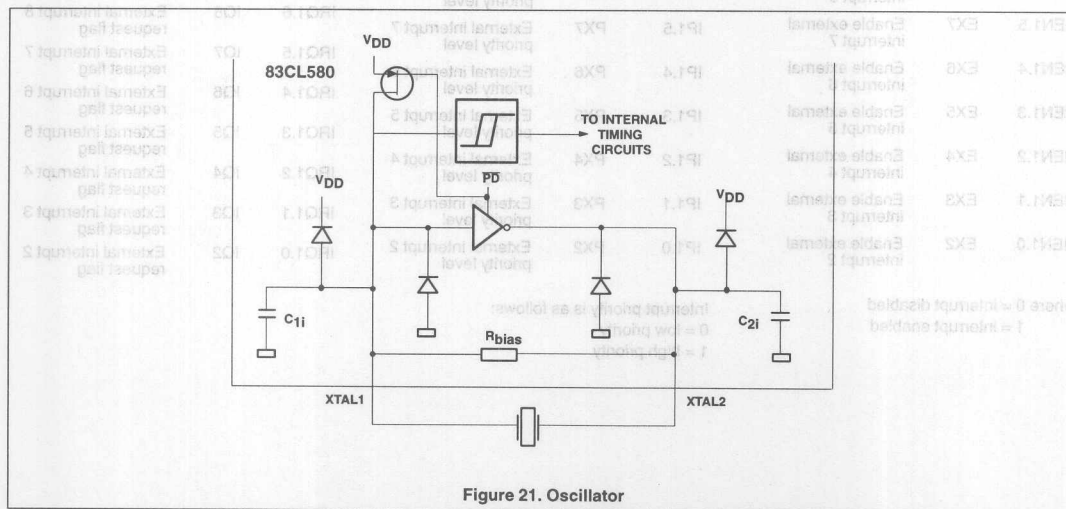
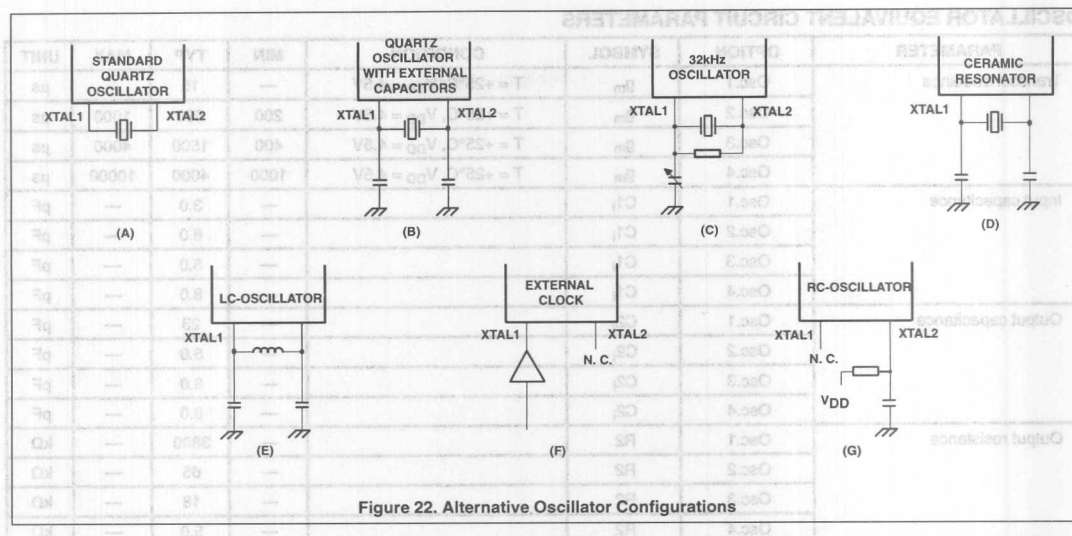


Figure 21. Oscillator

## Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580



## 1.8.2 Oscillator Type Selection Guide

RESONATOR	f (MHz)	OPTION	C1 EXT. (pF)		C2 EXT. (pF)		MAX. RESONATOR SERIES RESISTANCE 15kΩ <sup>1</sup>
			MIN	MAX	MIN	MAX	
Quartz	0.032	Osc. 1	0	0	5	15	600Ω
Quartz	1.0	Osc.2	0	30	0	30	100Ω
Quartz	3.58	Osc.2	0	15	0	15	75Ω
Quartz	4.0	Osc.2	0	20	0	20	60Ω
Quartz	6.0	Osc.3	0	10	0	10	60Ω
Quartz	10.0	Osc.4	0	15	0	15	40Ω
Quartz	12.0	Osc.4	0	10	0	10	20Ω
Quartz	16.0	Osc.4	0	15	0	15	10Ω
PXE	0.455	Osc.2	40	50	40	50	100Ω
PXE	1.0	Osc.2	15	50	15	50	10Ω
PXE	3.58	Osc.2	0	40	0	40	10Ω
PXE	4.0	Osc.2	0	40	0	40	5Ω
PXE	6.0	Osc.2	0	20	0	20	6Ω
PXE	10.0	Osc.3	0	15	0	15	6Ω
PXE	12.0	Osc.4	10	40	10	40	10μH = 1Ω
LC		Osc.2	20	90	20	90	100μH = 5Ω 1mH = 75Ω

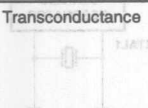
## NOTES:

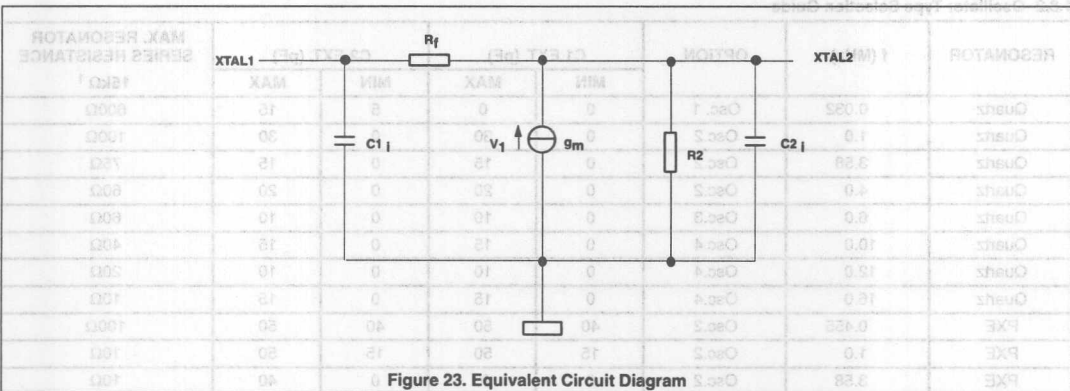
- 32kHz quartz crystals with a series resistance higher than 15kΩ will reduce the guaranteed supply voltage range to 2.5 - 3.5V.
- The equivalent circuit data of the internal oscillator compares with that of matched crystals.



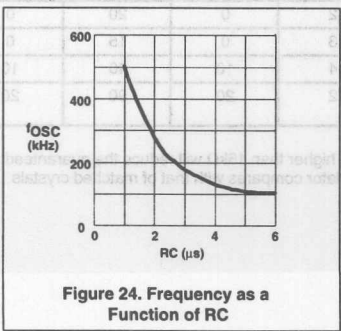
Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

OSCILLATOR EQUIVALENT CIRCUIT PARAMETERS

PARAMETER	OPTION	SYMBOL	CONDITION	MIN	TYP	MAX	UNIT
 Transconductance	Osc.1	$g_m$	$T = +25^{\circ}\text{C}, V_{DD} = 4.5\text{V}$	—	15	—	$\mu\text{S}$
	Osc.2	$g_m$	$T = +25^{\circ}\text{C}, V_{DD} = 4.5\text{V}$	200	600	1000	$\mu\text{S}$
	Osc.3	$g_m$	$T = +25^{\circ}\text{C}, V_{DD} = 4.5\text{V}$	400	1500	4000	$\mu\text{S}$
	Osc.4	$g_m$	$T = +25^{\circ}\text{C}, V_{DD} = 4.5\text{V}$	1000	4000	10000	$\mu\text{S}$
Input capacitance	Osc.1	$C_{1i}$		—	3.0	—	pF
	Osc.2	$C_{1i}$		—	8.0	—	pF
	Osc.3	$C_{1i}$		—	8.0	—	pF
	Osc.4	$C_{1i}$		—	8.0	—	pF
Output capacitance	Osc.1	$C_{2i}$		—	23	—	pF
	Osc.2	$C_{2i}$		—	8.0	—	pF
	Osc.3	$C_{2i}$		—	8.0	—	pF
	Osc.4	$C_{2i}$		—	8.0	—	pF
Output resistance	Osc.1	$R_2$		—	3800	—	k $\Omega$
	Osc.2	$R_2$		—	65	—	k $\Omega$
	Osc.3	$R_2$		—	18	—	k $\Omega$
	Osc.4	$R_2$		—	5.0	—	k $\Omega$



**1.8.3 RC Oscillator**  
The externally adjustable RC-oscillator has a frequency range from 100kHz to 500kHz.



## Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

### 1.9 Reset Circuitry

To initialize the 83CL580, a reset is performed by either of two methods:

- via the RST pin
- via a power-on-reset

It leaves the internal registers as follows:

Register	Content
ACC	0000 0000
ADCH	1111 1111
ADCON	X000 0000
B	0000 0000
DPL	0000 0000
DPH	0000 0000
IEN0	0000 0000
IEN1	0000 0000
IP0	X000 0000
IP1	0000 0000
IX1	0000 0000
IRQ1	0000 0000
PCH	0000 0000
PCL	0000 0000
PCON	0XX0 0000
P0 - P4	1111 1111
PWM0	0000 0000
PWMP	0000 0000
RCAP2H	0000 0000
RCAP2L	0000 0000
S0BUF	XXXXXXXX
S0CON	0000 0000
S1ADR	0000 0000
S1CON	0000 0000
S1DAT	0000 0000
S1STA	1111 1000
PSW	0000 0000
SP	0000 0111
TCON	0000 0000
T2CON	0000 0000
T3	0000 0000
TH0, TH1, TH2	0000 0000
TL0, TL1, TL2	0000 0000
TMOD	0000 0000

The reset state of the port pins is mask-programmable and can therefore be defined by the user. The standard reset value for port P0-P4 is 1111 1111.

The reset input to the 83CL580 is the RST pin. A Schmitt trigger qualifies the input for noise rejection. The output of the Schmitt trigger is sampled by the reset circuitry every machine cycle.

A reset is accomplished by holding the RST pin HIGH for at least two machine cycles (24 oscillator periods), while the oscillator is running. The CPU responds by generating an internal reset. Port pins adopt their reset state immediately after RST goes HIGH. During reset ALE and PSEN are held HIGH.

The external reset is asynchronous to the internal clock. The RST pin is sampled during State 5, Phase 2 of every machine cycle. After a HIGH is detected at the RST pin, an internal reset is repeated every cycle until RST goes LOW. The reset circuitry is also affected by the watchdog timer T3, see paragraph 1.3.2.

The internal RAM is not affected by reset. When  $V_{DD}$  is turned on the RAM contents are indeterminate.

#### 1.9.1 Power-on Reset

The 83CL580 contains on-chip circuitry which switch the port pins to the customer defined logic level as soon as  $V_{DD}$  exceeds 1.3V if

the mask option "ON" has been chosen. As soon as the minimum supply voltage is reached, the oscillator will start up. However, to ensure that the oscillator is stable before the controller starts, the clock signals are gated away from the CPU for a further 1536 oscillator periods. During that time the CPU is held in a reset state.

A hysteresis of approximately 50mV at a typical power-on switching level of 1.3V will ensure correct operation.

The on-chip Power-on reset circuitry can also be switched off via the mask option "OFF". This option reduces the Power-down current to typically 800nA and can be chosen if external reset circuitry is used. For applications not requiring the internal reset, option 'OFF' should be chosen.

An automatic reset can be obtained at power-on by connecting the RST pin to  $V_{DD}$  via a 10 $\mu$ F capacitor. At power-on, the voltage on the RST pin is equal to  $V_{DD}$  minus the capacitor voltage, and decreases from  $V_{DD}$  as the capacitor discharges through the internal resistor  $R_{RST}$  to ground. The larger the capacitor, the more slowly  $V_{RST}$  decreases.  $V_{RST}$  must remain above the lower threshold of the Schmitt trigger long enough to effect a complete reset. The time required is the oscillator start-up time, plus 2 machine cycles.

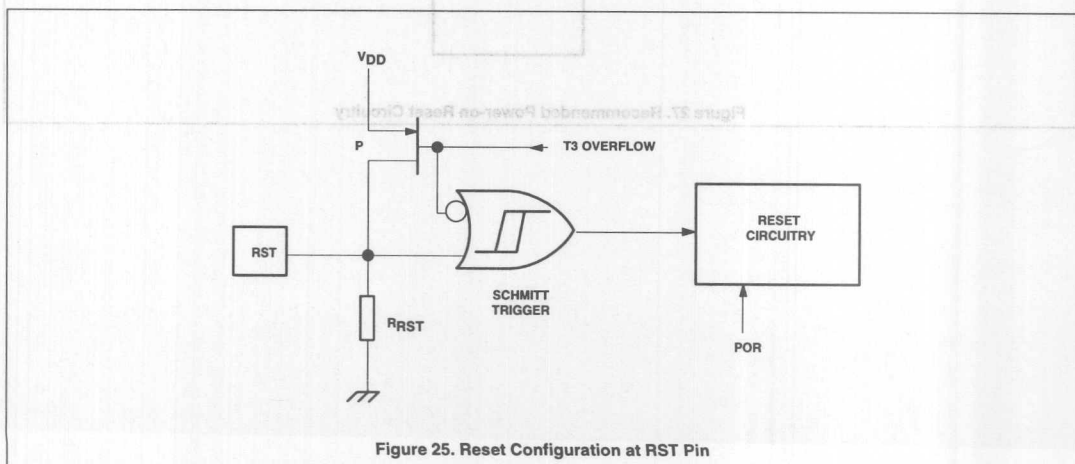
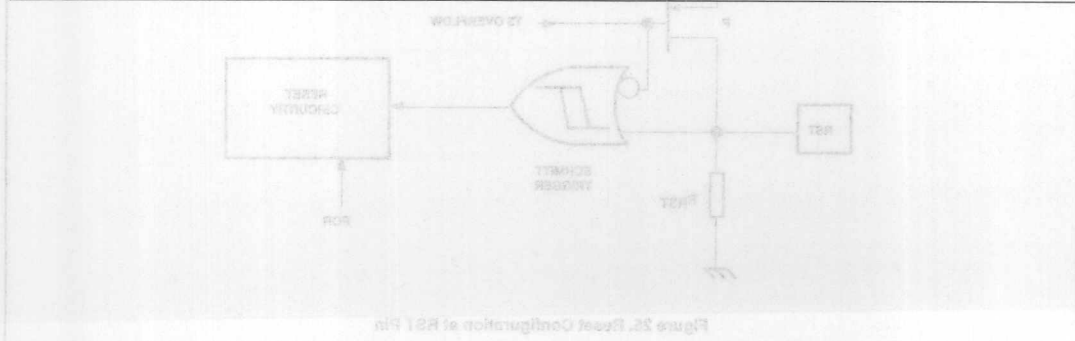
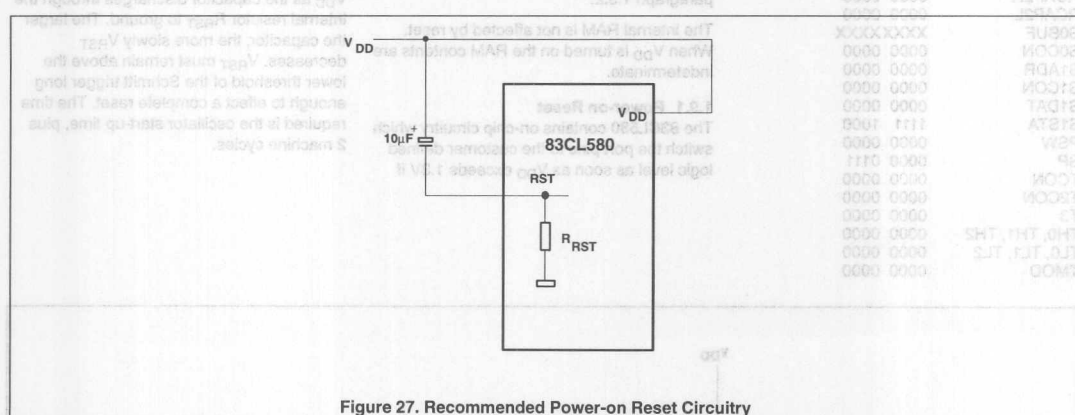
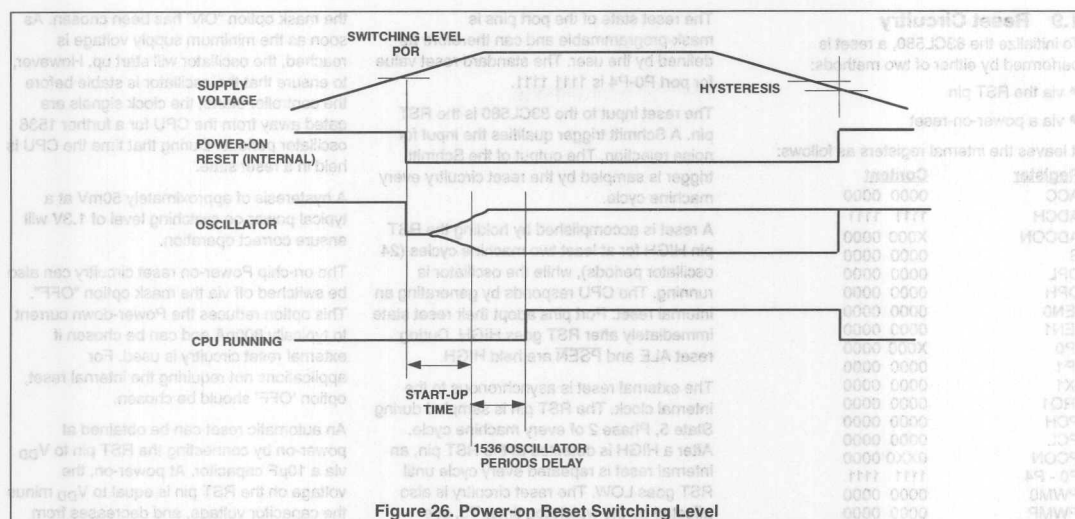
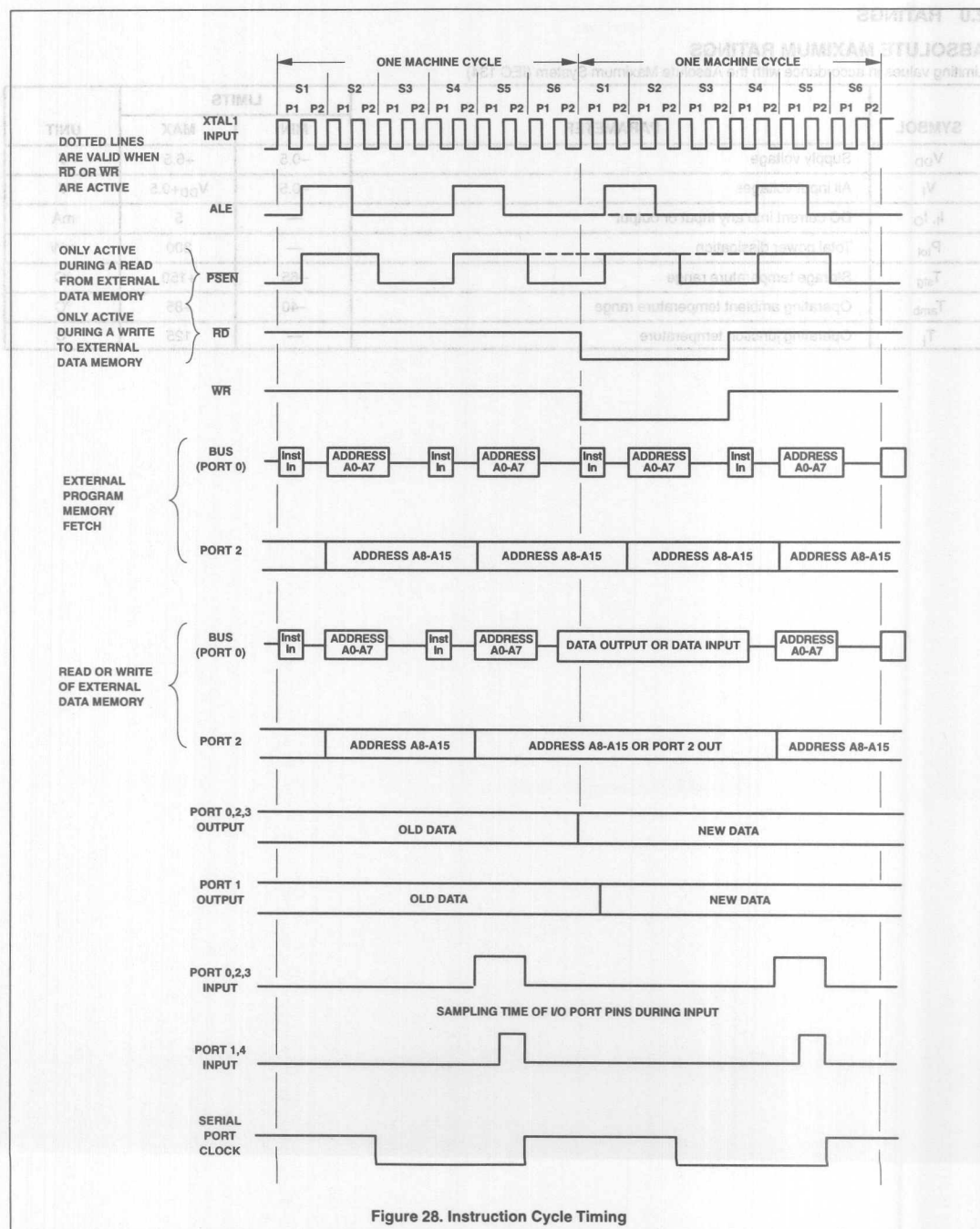


Figure 25. Reset Configuration at RST Pin

## Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580



Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580



Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

## 2.0 RATINGS

### ABSOLUTE MAXIMUM RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

SYMBOL	PARAMETER	LIMITS		UNIT
		MIN	MAX	
V <sub>DD</sub>	Supply voltage	-0.5	+6.5	V
V <sub>I</sub>	All input voltages	-0.5	V <sub>DD</sub> +0.5	V
I <sub>I</sub> , I <sub>O</sub>	DC current into any input or output	—	5	mA
P <sub>tot</sub>	Total power dissipation	—	300	mW
T <sub>stg</sub>	Storage temperature range	-65	+150	°C
T <sub>amb</sub>	Operating ambient temperature range	-40	+85	°C
T <sub>J</sub>	Operating junction temperature	—	125	°C



# Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

## 3.0 DC ELECTRICAL CHARACTERISTICS

$V_{DD} = 2.5V$  to  $6V$ ;  $V_{SS} = 0V$ ;  $T_{amb} = -40^{\circ}C$  to  $+85^{\circ}C$ , all voltages with respect to  $V_{SS}$ , unless otherwise specified.

SYMBOL		PARAMETER	TEST CONDITIONS	LIMITS			UNIT
				MIN	TYP	MAX	
Supply voltage							
V <sub>DD</sub>	Supply voltage		2.5	—	6.0	V	
	RAM retention voltage in power-down mode		1.0	—	6.0	V	
Supply current							
I <sub>DD</sub>	Operating Supply current (note 1)	V <sub>DD</sub> = 5V; f <sub>clk</sub> = 12 MHz	—	—	27.0	mA	
		V <sub>DD</sub> = 3V; f <sub>clk</sub> = 3.58 MHz	—	—	5.0	mA	
	Idle mode Supply current (note 2)	V <sub>DD</sub> = 5V; f <sub>clk</sub> = 12 MHz	—	—	10.0	mA	
		V <sub>DD</sub> = 3V; f <sub>clk</sub> = 3.58 MHz	—	—	3.0	mA	
I <sub>PD</sub>	Power-down mode (note 3)	V <sub>DD</sub> = 1.8V; T <sub>amb</sub> = 25 °C	—	—	10	μA	
Inputs (Note 6)							
V <sub>IL</sub>	Input voltage LOW		V <sub>SS</sub>	—	0.3V <sub>DD</sub>	V	
V <sub>IH</sub>	Input voltage HIGH		0.7V <sub>DD</sub>	—	V <sub>DD</sub>	V	
±I <sub>LI</sub>	Input leakage current ( port 0, EA)	V <sub>SS</sub> < V <sub>I</sub> < V <sub>DD</sub>	—	—	10	μA	
Outputs							
I <sub>OL</sub>	Output sink current LOW (except SDA, SCL)	V <sub>DD</sub> = 5V; V <sub>OL</sub> = 0.4V	1.6	—	—	mA	
		V <sub>DD</sub> = 2.5V; V <sub>OL</sub> = 0.4V	0.7	—	—	mA	
	Output sink current, SDA, SCL	V <sub>DD</sub> = 5V; V <sub>OL</sub> = 0.4V	3.0	—	—	mA	
		V <sub>DD</sub> = 5V; V <sub>OL</sub> = 0.4V	3.2	—	—	mA	
		V <sub>DD</sub> = 2.5V; V <sub>OL</sub> = 0.4V	1.6	—	—	mA	
-I <sub>OH</sub>	PWM0	V <sub>DD</sub> = 5V; V <sub>OH</sub> = V <sub>DD</sub> -0.4V	3.2	—	—	mA	
		V <sub>DD</sub> = 2.5V; V <sub>OH</sub> = V <sub>DD</sub> -0.4V	1.6	—	—	mA	
	Output source current HIGH, push-pull options only	V <sub>DD</sub> = 5V; V <sub>OH</sub> = V <sub>DD</sub> -0.4V	1.6	—	—	mA	
		V <sub>DD</sub> = 2.5V; V <sub>OH</sub> = V <sub>DD</sub> -0.4V	0.7	—	—	mA	
-I <sub>IL</sub>	Input current logic 0	V <sub>DD</sub> = 5V; V <sub>IN</sub> = 0.4V	—	—	100	μA	
		V <sub>DD</sub> = 2.5V; V <sub>IN</sub> = 0.4V	—	—	50	μA	
-I <sub>TL</sub>	Input current logic 0, 1-to-0 transition	V <sub>DD</sub> = 5V; V <sub>IN</sub> = V <sub>DD</sub> /2	—	—	1.0	mA	
		V <sub>DD</sub> = 2.5V; V <sub>IN</sub> = V <sub>DD</sub> /2	—	—	500	μA	
R <sub>RST</sub>	RST pull-down resistor		10	—	200	kΩ	

$V_{DD} = 2.5V$  to  $6V$ ;  $V_{SS} = 0V$ ;  $AV_{SS} = 0V$ ;  $AV_{REFP} = V_{DD}$ ;  $T_{amb} = -40^{\circ}C$  to  $+85^{\circ}C$ , unless otherwise specified.

### Analogue Inputs

$AV_{IN}$	Analog input voltage		$AV_{SS}$	—	$AV_{DD}$	V
$AV_{REFP}$	Reference voltage		2.5	—	$V_{DD}$	V
$R_{REF}$	Resistance between $AV_{REFP}$ and $AV_{SS}$		25	—	100	kΩ
$C_{AIN}$	Analog on-chip input capacitance		—	3	—	pF
	Absolute error (note 7):	$\pm 1\text{ LSB}$				
	Zero-offset error (note 8):	$\pm 1\text{ LSB}$				
	Differential non-linearity (note 9):	$\pm 1\text{ LSB}$				
	Channel to channel variation (note 10):	$\pm 1/2\text{ LSB}$				

NOTES ON NEXT PAGE.

# Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

## NOTES TO DC CHARACTERISTICS:

1. The operating supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 10\text{ns}$ ;  $V_{IL} = V_{SS}$ ;  $V_{IH} = V_{DD}$ ; XTAL1 not connected; EA = RST + PORT 0 =  $V_{DD}$ .
2. The idle mode supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 10\text{ns}$ ;  $V_{IL} = V_{SS}$ ;  $V_{IH} = V_{DD}$ ; XTAL2 is not connected; EA = PORT 0 =  $V_{DD}$ .
3. The power-down current is measured with all output pins disconnected; XTAL1 not connected; EA = Port 0 =  $V_{DD}$ ; RST =  $V_{SS}$ .
4. Capacitive loading on Port 0 and Port 2 may cause spurious noise pulses to be superimposed on the LOW level output voltage of ALE, Port 1 and Port 3 pins when these pins make a 1-to-0 transition during bus operations. In the most adverse conditions (capacitive loading > 100pF) the noise pulse on the ALE line may exceed 0.8V. In this event it may be required to qualify ALE with a Schmitt trigger, or use an address latch with a Schmitt trigger strobe input.
5. Capacitive loading on Port 0 and Port 2 may cause the HIGH level output voltage on ALE and PSEN to momentarily fall below the 0.9% of  $V_{DD}$  specification when the address bits are stabilizing.
6. The input threshold voltage of P1.6/SCL and P1.7/SDA meet the I<sup>2</sup>C-bus specification, so an input voltage below  $0.3V_{DD}$  will be recognized as a logic 0, while an input voltage above  $0.7V_{DD}$  will be recognized as a logic 1.
7. Absolute error — The maximum difference between actual and ideal code transitions. Absolute error accounts for all deviations of an actual converter from an ideal converter.
8. Zero offset — The difference between the actual and ideal input voltage corresponding to the first actual code transition.
9. Differential non-linearity — The difference between the actual and ideal code widths.
10. Channel to channel variation — The difference between corresponding code transitions of actual characteristics taken from different channels under the same temperature, voltage and frequency conditions. Not tested, but verified on sampling basis.

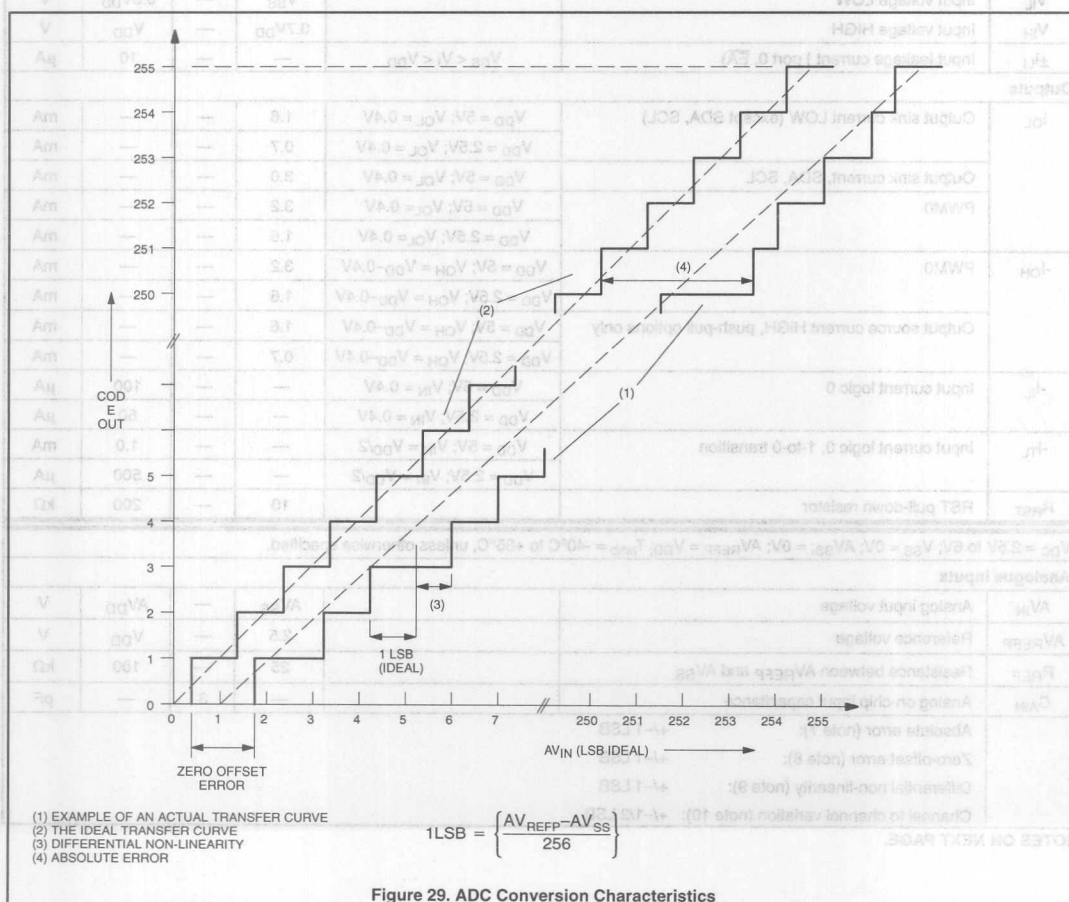


Figure 29. ADC Conversion Characteristics

# Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

## 4.0 AC CHARACTERISTICS

### AC ELECTRICAL CHARACTERISTICS

$V_{DD} = 5V$ ;  $V_{SS} = 0V$ ;  $T_{amb} = -40$  to  $+85^{\circ}C$ ;  $C_L = 50pF$  for Port 0, ALE and PSEN;  $C_L = 40pF$  for all other outputs unless otherwise specified.

SYMBOL	PARAMETER	12MHz CLOCK		VARIABLE CLOCK		UNIT
		MIN	MAX	MIN	MAX	
Program Memory (see Figure 30)						
t <sub>LL</sub>	ALE pulse duration	127	—	2t <sub>CK</sub> -40	—	ns
t <sub>AL</sub>	Address set-up time to ALE	43	—	t <sub>CK</sub> -40	—	ns
t <sub>LA</sub>	Address hold time after ALE	48	—	t <sub>CK</sub> -35	—	ns
t <sub>LC</sub>	Time from ALE to control pulse PSEN	58	—	t <sub>CK</sub> -25	—	ns
t <sub>LIV</sub>	Time from ALE to valid instruction input	—	233	—	4t <sub>CK</sub> -100	ns
t <sub>CC</sub>	Control pulse duration PSEN	215	—	3t <sub>CK</sub> -35	—	ns
t <sub>CIV</sub>	Time from PSEN to valid instruction input	—	125	—	3t <sub>CK</sub> -125	ns
t <sub>CI</sub>	Input instruction hold time after PSEN	0	—	0	—	ns
t <sub>CIF</sub>	Input instruction float delay after PSEN	—	63	—	t <sub>CK</sub> -20	ns
t <sub>AC</sub>	Address valid after PSEN	75	—	t <sub>CK</sub> -8	—	ns
t <sub>AIV</sub>	Address to valid instruction input	—	302	—	5t <sub>CK</sub> -115	ns
t <sub>AFC</sub>	Address float time to PSEN	12	—	0	—	ns

### AC ELECTRICAL CHARACTERISTICS

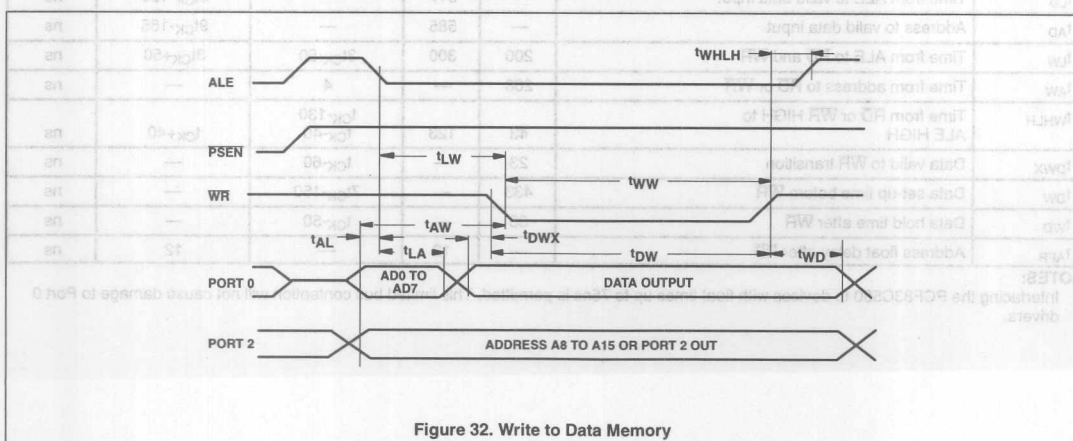
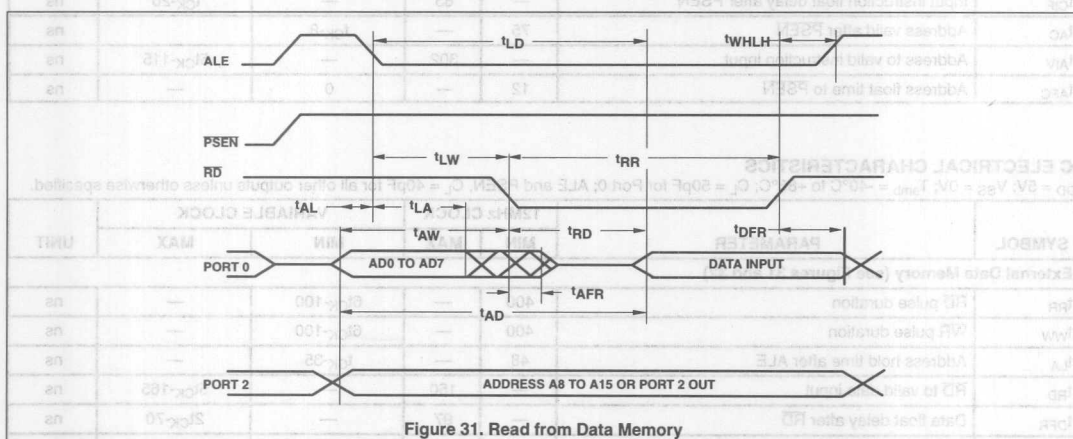
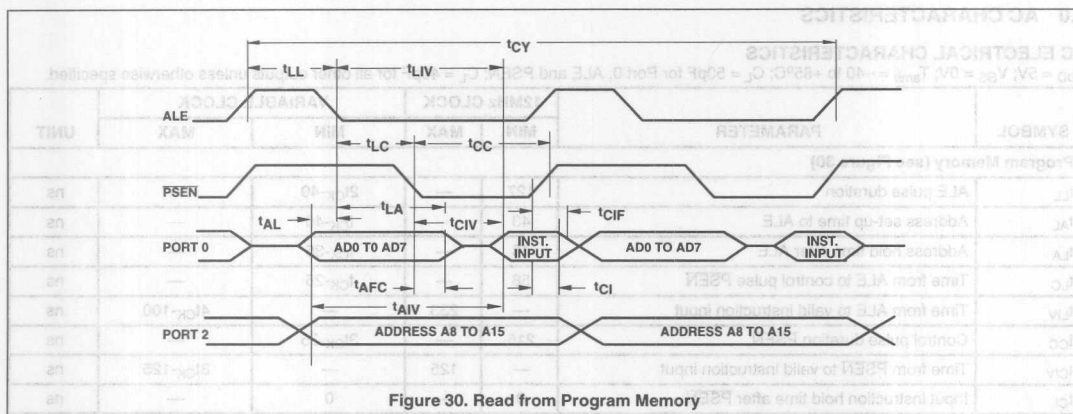
$V_{DD} = 5V$ ;  $V_{SS} = 0V$ ;  $T_{amb} = -40^{\circ}C$  to  $+85^{\circ}C$ ;  $C_L = 50pF$  for Port 0; ALE and PSEN,  $C_L = 40pF$  for all other outputs unless otherwise specified.

SYMBOL	PARAMETER	12MHz CLOCK		VARIABLE CLOCK		UNIT
		MIN	MAX	MIN	MAX	
External Data Memory (see Figures 31 and 32)						
t <sub>RR</sub>	R <sub>D</sub> pulse duration	400	—	6t <sub>CK</sub> -100	—	ns
t <sub>WW</sub>	W <sub>R</sub> pulse duration	400	—	6t <sub>CK</sub> -100	—	ns
t <sub>LA</sub>	Address hold time after ALE	48	—	t <sub>CK</sub> -35	—	ns
t <sub>RD</sub>	R <sub>D</sub> to valid data input	—	150	—	5t <sub>CK</sub> -165	ns
t <sub>DFR</sub>	Data float delay after R <sub>D</sub>	—	97	—	2t <sub>CK</sub> -70	ns
t <sub>LD</sub>	Time from ALE to valid data input	—	517	—	8t <sub>CK</sub> -150	ns
t <sub>AD</sub>	Address to valid data input	—	585	—	9t <sub>CK</sub> -165	ns
t <sub>LW</sub>	Time from ALE to R <sub>D</sub> and W <sub>R</sub>	200	300	3t <sub>CK</sub> -50	3t <sub>CK</sub> +50	ns
t <sub>AW</sub>	Time from address to R <sub>D</sub> or W <sub>R</sub>	203	—	4	—	ns
t <sub>WHLH</sub>	Time from R <sub>D</sub> or W <sub>R</sub> HIGH to ALE HIGH	43	123	t <sub>CK</sub> -130 t <sub>CK</sub> -40	t <sub>CK</sub> +40	ns
t <sub>DWX</sub>	Data valid to W <sub>R</sub> transition	23	—	t <sub>CK</sub> -60	—	ns
t <sub>DW</sub>	Data set-up time before W <sub>R</sub>	433	—	7t <sub>CK</sub> -150	—	ns
t <sub>WD</sub>	Data hold time after W <sub>R</sub>	33	—	t <sub>CK</sub> -50	—	ns
t <sub>AFR</sub>	Address float delay after R <sub>D</sub>	—	12	—	12	ns

#### NOTES:

- Interfacing the PCF83C580 to devices with float times up to 75ns is permitted. This limited bus contention will not cause damage to Port 0 drivers.

## Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580



## Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580

## 5.0 CHARACTERISTIC CURVES

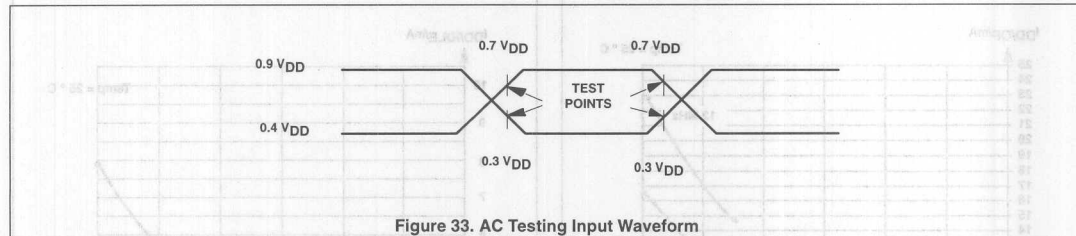


Figure 33. AC Testing Input Waveform

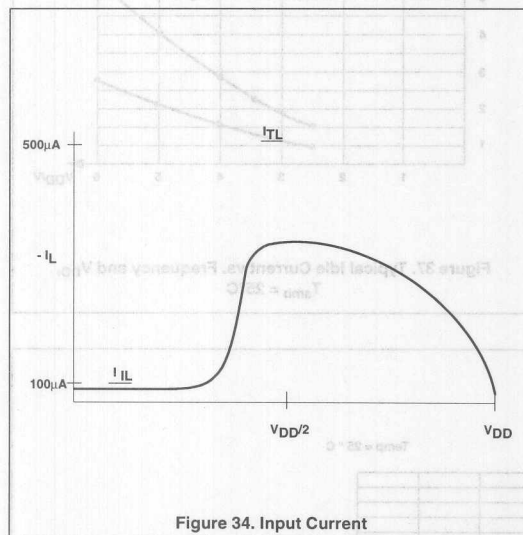


Figure 34. Input Current

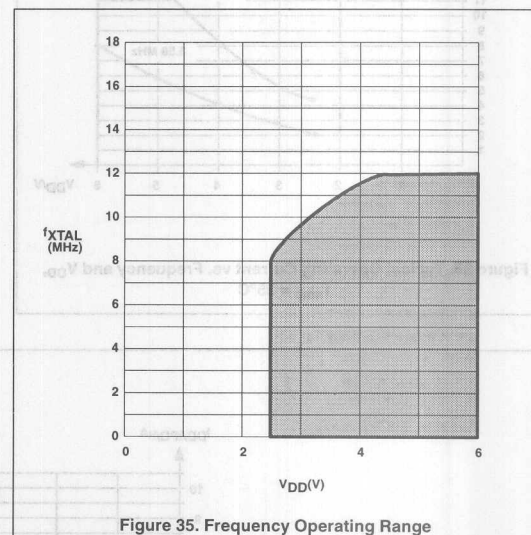
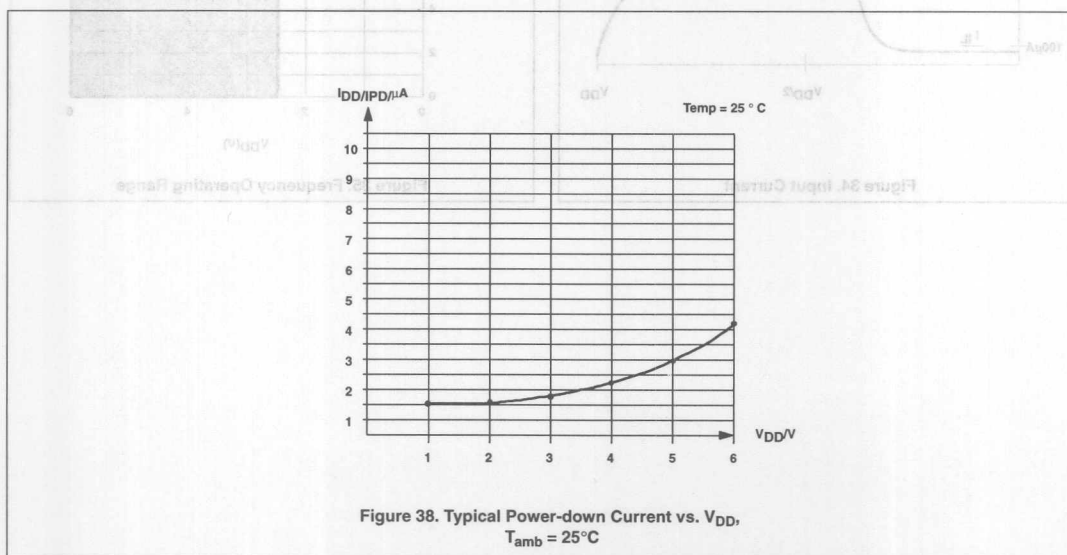
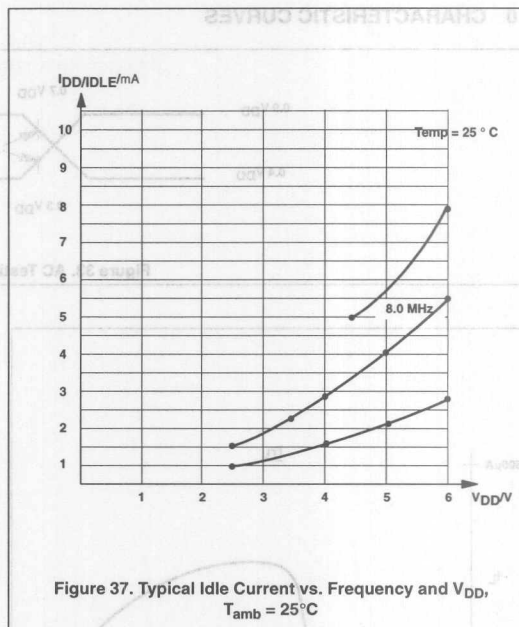
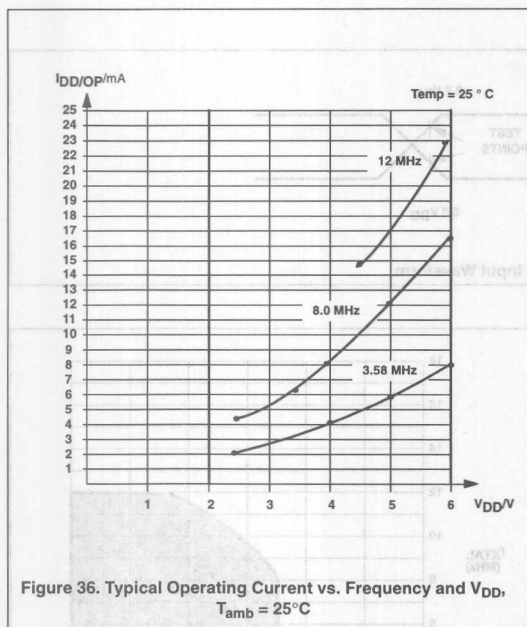


Figure 35. Frequency Operating Range



## Low-voltage single-chip 8-bit microcontroller 80CL580/83CL580



## Single-chip 8-bit microcontroller with CAN controller

**80C592/83C592/87C592**

### DESCRIPTION

The 80C592/83C592/87C592 (hereafter referred to generically as the 8XC592) is a stand-alone high-performance microcontroller designed for use in automotive and general industrial applications. In addition to the 80C51 standard features, this device provides a number of dedicated hardware functions for these applications. Three versions of this derivative will be offered:

- 83C592 (ROM version)
- 80C592 (ROMless version)
- 87C592 (EPROM/OTP version)

It combines the functions of the existing 8XC552 and the Philips CAN-Controller PCA82C200 (CAN: Controller Area Network) with the following enhanced features:

- 16K byte Program Memory
- 2 × 256 byte Data Memory
- DMA between CAN Transmit/Receive buffer and internal RAM

The temperature range includes -40°C to +85°C as well as automotive temperature range -40°C to +125°C for the ROM and ROMless version with a maximum clock frequency of 16MHz. The 87C592 has a temperature range of -40°C to +85°C.

The main differences to the 8XC552 microcontroller are:

- a CAN-controller substitutes the I<sup>2</sup>C-serial interface
- 16K byte programmable ROM resp. EPROM instead of 8K byte
- additional 256 byte RAM.

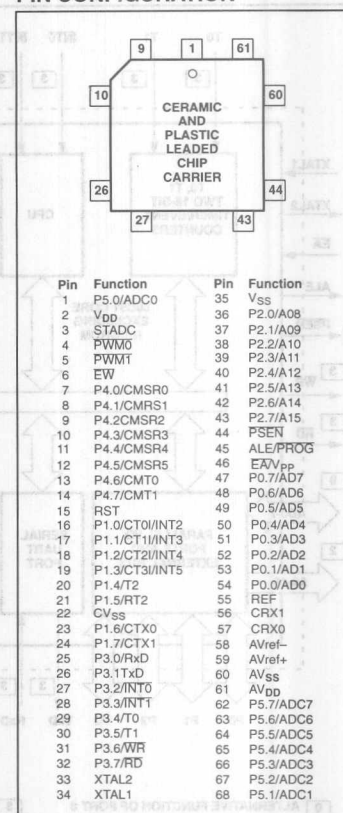
The 8XC592 contains a 16k × 8 EPROM (87C592), ROM (83C592) program memory, a volatile 512 × 8 read/write data memory, a Controller Area Network (CAN) controller, six 8-bit I/O ports, one 8-bit input port, two 16-bit timer/event counters (identical to the timers of the 80C51), an additional 16-bit timer coupled to capture and compare latches, a 15-source, two-priority-level, nested interrupt structure, a 10-input ADC, a dual DAC pulse width modulated interface, two serial interfaces

(UART and CAN), a "watchdog" timer and on-chip oscillator and timing circuits. For systems that require extra capability, the 8XC592 memory can be expanded externally using standard TTL compatible memories and logic.

### FEATURES

- 80C51 core architecture
- 16k × 8 EPROM (87C592)
- 16k × 8 ROM (83C592)
- ROMless (80C592)
- 512 × 8 RAM, expandable externally to 64k bytes
- Two standard 16-bit timer/counters
- An additional 16-bit timer/counter coupled to four capture registers and three compare registers
- A 10-bit ADC with eight multiplexed analog inputs
- Two 8-bit resolution, pulse width modulation outputs
- 15 interrupt sources with 2 priority levels
- Five 8-bit I/O ports plus one 8-bit input port shared with analog inputs
- CAN controller with DMA transfer between internal data RAM and CAN registers
- Up to 1 Mbit/s CAN-Controller with bus failure management facility
- V<sub>DD</sub>/2 reference voltage
- Full-duplex UART compatible with the standard 80C51
- On-chip watchdog timer
- Extended temperature ranges (-40 to +125°C)
- OTP package available
- ROM code protection

### PIN CONFIGURATION



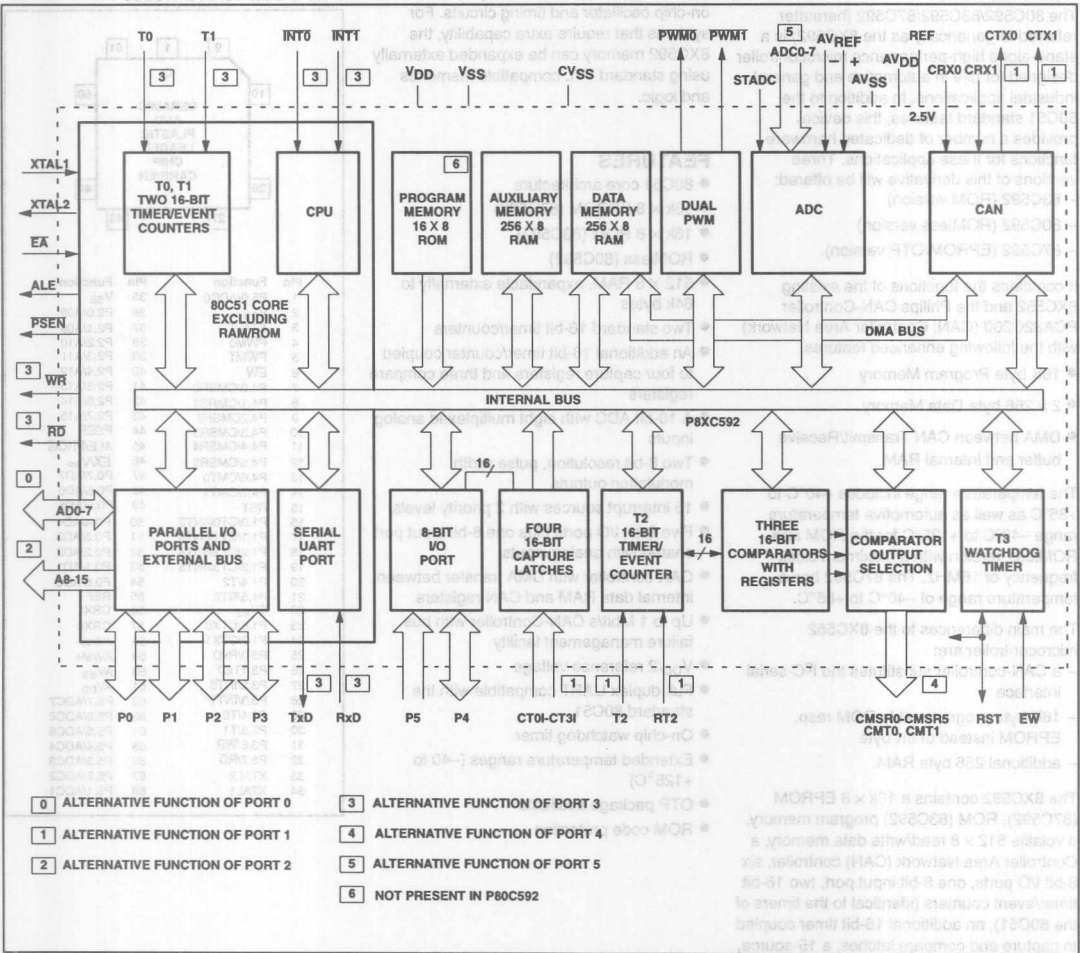
### ORDERING INFORMATION

ROMless	ROM	EPROM	TEMPERATURE RANGE °C AND PACKAGE	FREQUENCY MHZ	DRAWING NUMBER
P80C592FFA	P83C592FFA	—	-40 to +85, 68-Pin Plastic Leaded Chip Carrier	1.2 to 16	0398
P80C592FHA	P83C592FHA	—	-40 to +125, 68-Pin Plastic Leaded Chip Carrier	1.2 to 16	0398
—	—	P87C592EFL	-40 to +85, 68-Pin Ceramic Leaded Chip Carrier w/Window	3.5 to 16	1240
—	—	P87C592EFA	-40 to +85, 68-Pin Plastic Leaded Chip Carrier	3.5 to 16	0398

Single-chip 8-bit microcontroller  
with CAN controller

80C592/83C592/87C592

BLOCK DIAGRAM



DRAWING NUMBER	FREQUENCY MHz	TEMPERATURE RANGE °C AND PACKAGE	EPROM	ROM	ROMless
0088	1.5 to 16	-40 to +85, 68-Pin Plastic Leaded Chip Carrier	-	P80C592FA	P80C592FA
0088	1.5 to 16	-40 to +125, 68-Pin Plastic Leaded Chip Carrier	-	P80C592FAH	P80C592FAH
1240	3.5 to 16	-40 to +85, 68-Pin Ceramic Leaded Chip Carrier	P87C592EL	-	-
0088	3.5 to 16	-40 to +85, 68-Pin Plastic Leaded Chip Carrier	P87C592FA	-	-

# Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

## PIN DESCRIPTION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
V <sub>DD</sub>	2	I	<b>Digital Power Supply:</b> +5V power supply pin during normal operation, idle and power-down mode.
STADC	3	I	<b>Start ADC Operation:</b> Input starting analog to digital conversion (ADC operation can also be started by software). This pin must not float.
PWM0	4	O	<b>Pulse Width Modulation:</b> Output 0.
PWM1	5	O	<b>Pulse Width Modulation:</b> Output 1.
EW	6	I	<b>Enable Watchdog Timer:</b> Enable for T3 watchdog timer and disable power-down mode. This pin must not float.
P0.0-P0.7	54-47	I/O	<b>Port 0:</b> Port 0 is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application it uses strong internal pull-ups when emitting 1s. Port 0 is also used to input the code byte during programming and to output the code byte during verification.
P1.0-P1.7	16-21, 23-24	I/O	<b>Port 1:</b> 8-bit I/O port. Alternate functions include:
	16-19	I	<b>CT0-CT3 (P1.0-P1.3):</b> Capture timer input signals for timer T2.
	16-19	I	<b>INT2-INT5 (P1.0-P1.3):</b> External interrupts 2-5.
	20	I	<b>T2 (P1.4):</b> T2 event input. Rising edge triggered.
	21	I	<b>RT2 (P1.5):</b> T2 timer reset signal. Rising edge triggered.
	23	O	<b>CTX0 (P1.6):</b> CAN transmitter output 0.
	24	O	<b>CTX1 (P1.7):</b> CAN transmitter output 1.
			Port 1 is also used to input the lower order address byte during EPROM programming and verification. A0 is on P1.0, etc.
CV <sub>SS</sub>	22	I	<b>CV<sub>SS</sub>:</b> CAN transmitter driver ground.
P2.0-P2.7	36-43	I/O	<b>Port 2:</b> 8-bit quasi-bidirectional I/O port. Alternate function: High-order address byte for external memory (A08-A15). Port 2 is also used to input the upper order address during EPROM programming and verification. A8 is on P2.0, A9 on P2.1, through A12 on P2.4.
P3.0-P3.7	25-32	I/O	<b>Port 3:</b> 8-bit quasi-bidirectional I/O port. Alternate functions include:
	25		<b>RxD (P3.0):</b> Serial input port.
	26		<b>TxD (P3.1):</b> Serial output port.
	27		<b>INT0 (P3.2):</b> External interrupt.
	28		<b>INT1 (P3.3):</b> External interrupt.
	29		<b>T0 (P3.4):</b> Timer 0 external input.
	30		<b>T1 (P3.5):</b> Timer 1 external input.
	31		<b>WR (P3.6):</b> External data memory write strobe.
	32		<b>RD (P3.7):</b> External data memory read strobe.
P4.0-P4.7	7-14	I/O	<b>Port 4:</b> 8-bit quasi-bidirectional I/O port. Alternate functions include:
	7-12	O	<b>CMSR0-CMSR5 (P4.0-P4.5):</b> Timer T2 compare and set/reset outputs on a match with timer T2.
	13, 14	O	<b>CMT0, CMT1 (P4.6, P4.7):</b> Timer T2 compare and toggle outputs on a match with timer T2.
P5.0-P5.7	68-62, 1	I	<b>Port 5:</b> 8-bit input port. <b>ADC0-ADC7 (P5.0-P5.7):</b> Alternate function: Eight input channels to ADC.
RST	15	I/O	<b>Reset:</b> Input to reset the 8XC592. It also provides a reset pulse as output when the watchdog timer overflows or after a CAN wakeup from power-down.
XTAL1	34	I	<b>Crystal Pin 1:</b> Input to the inverting amplifier that forms the oscillator, and input to the internal clock generator. Receives the external clock signal when an external oscillator is used.
XTAL2	33	O	<b>Crystal Pin 2:</b> Output of the inverting amplifier that forms the oscillator. Left open-circuit when an external clock is used.
V <sub>SS</sub>	35	I	<b>Digital ground.</b>
PSEN	44	O	<b>Program Store Enable:</b> Active-low read strobe to external program memory.
ALE/PROG	45	O	<b>Address Latch Enable:</b> Latches the low byte of the address during accesses to external memory. It is activated every six oscillator periods. During an external data memory access, one ALE pulse is skipped. ALE can drive up to eight LS TTL inputs and handles CMOS inputs without an external pull-up. This pin is also the program pulse input (PROG) during EPROM programming.

# Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

## PIN DESCRIPTION (Continued)

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
EA/V <sub>PP</sub>	46	I	<b>External Access:</b> When EA is held at TTL level high, during reset the CPU executes out of the internal program ROM provided the program counter is less than 16384. When EA is held at TTL low level, during reset the CPU executes out of external program memory. EA is not allowed to float. This pin also receives the 12.75V programming supply voltage (V <sub>PP</sub> ) during EPROM programming.
REF	55	O	<b>REF:</b> AV <sub>DD</sub> /2 reference voltage output or input, depending on CAN control register bits. If the internal reference is used, then REF should be connected to AV <sub>SS</sub> through a 10nF (or greater) capacitor.
CRX1	56	I	<b>CRX1:</b> CAN receiver input line 1.
CRX0	57	I	<b>CRX0:</b> CAN receiver input line 0.
AV <sub>REF-</sub>	58	I	<b>Analog to Digital Conversion Reference Resistor:</b> Low-end.
AV <sub>REF+</sub>	59	I	<b>Analog to Digital Conversion Reference Resistor:</b> High-end.
AV <sub>SS</sub>	60	I	<b>Analog Ground</b> (for ADC and CAN receiver)
AV <sub>DD</sub>	61	I	<b>Analog Power Supply</b> (for ADC and CAN receiver)

### NOTE:

1. To avoid "latch-up" effect at power-on, the voltage on any pin at any time must not be higher or lower than V<sub>DD</sub> + 0.5V or V<sub>SS</sub> - 0.5V, respectively.

The 8XC592 has the same operation as the 8XC552 for all features except the CAN interface and the AuxRAM. Please refer to the 8XC552 section in this data handbook for information on the PWM outputs, A/D converter, Timers 0, 1, or 2, the Watchdog Timer and the UART (SIO0).

## INTERNAL DATA MEMORY

The internal Data Memory is divided into three physically separated parts: the 256 byte of Main RAM, the 256 byte of AuxRAM, and

the 128 byte special function area. these can be addressed each in a different way.

- Main RAM 0 to 127 can be addressed directly and indirectly as in the 80C51. Address pointers are R0 and R1 of the selected register bank.
- Main RAM 128 to 255 can only be addressed indirectly. Address pointers are R0 and R1 of the selected register bank.
- AuxRAM 0 to 255 is indirectly addressable in the same way as the external Data Memory with MOVX instructions. Address pointers are R0, R1 of the selected register

bank and the DPTR. An access to AuxRAM 0 to 255 will not affect the ports P0, P2, P3.6 and P3.7 during internal program execution.

An access to external Data Memory locations higher than 255 will be performed with the MOVX @DPTR instructions in the same way as in the 80C51 structure, so with P0 and P2 as data/address bus and P3.6 and P3.7 as write and read strobe signals. Note that these external Data Memory locations cannot be accessed with R0 or R1 as address pointer.

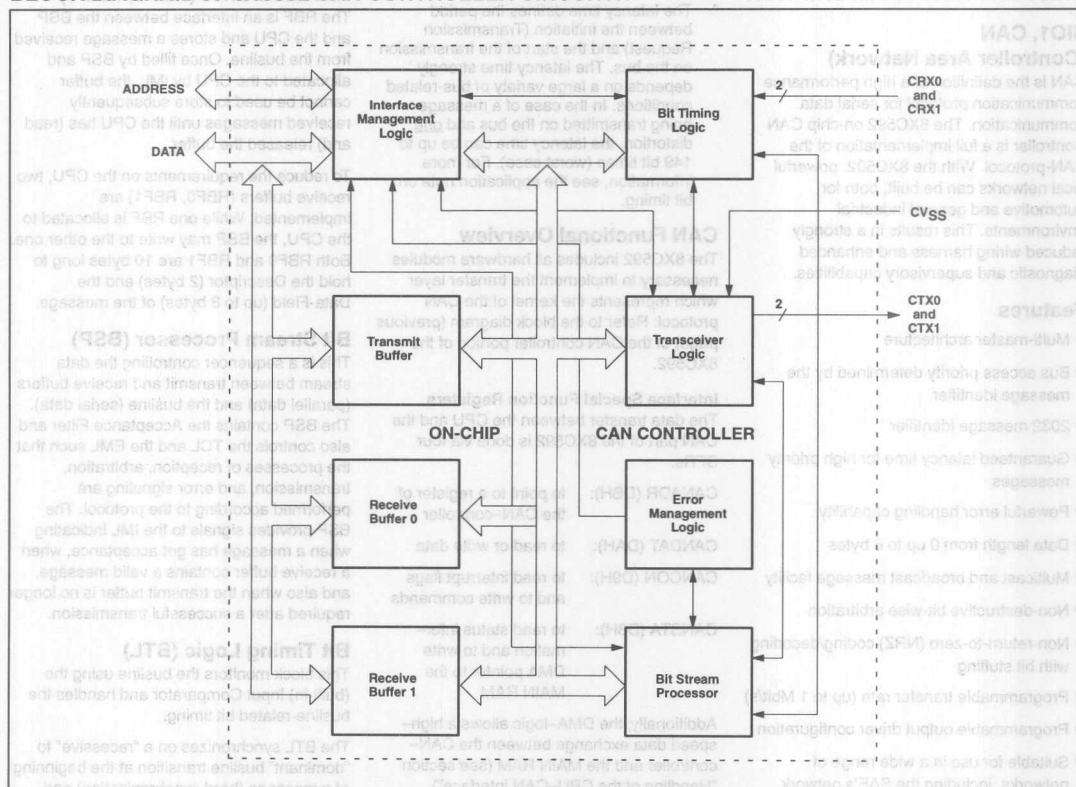
WDT (P3.2): External interrupt.	37		
WDT (P3.3): External interrupt.	38		
TD (P3.4): Timer 0 external input.	39		
T1 (P3.5): Timer 1 external input.	40		
WR (P3.6): External data memory write strobe.	41		
RD (P3.7): External data memory read strobe.	42		
Port 4: 8-bit quasi-bidirectional I/O port. Alternate functions include: CM2RD-CM2RS (P4.0-P4.5): Timer T2 compare and toggle outputs on a match with timer T2.	43		
CM2G-CM2T (P4.6, P4.7): Timer T2 compare and toggle outputs on a match with timer T2.	44		
Port 5: 8-bit input port.	45		
ADDA-ADCF (P5.0-P5.7): Alternate function. Eight input channels to ADC.	46		
Reset input to reset the 8XC592. It also provides a read pulse as output when the watchdog timer overflows or after a CAN wakeup from power-down.	47		
Crystal Pin 1: Input to the inverting amplifier that forms the oscillator, and input to the internal clock generator. Receives the external clock signal when an external oscillator is used.	48		
Crystal Pin 2: Output of the inverting amplifier that forms the oscillator. Left open-circuit when an external clock is used.	49		
Digital ground.	50		
Program Store Enable: Active-low read strobe to external program memory.	51		
Address Latch Enable: Latches the low byte of the address during accesses to external memory. It is activated every six oscillator periods. During an external data memory access, one ALE pulse is shipped. ALE can drive up to eight LS TTL inputs and handles CMOS inputs without an external pull-up. This pin is also the program pulse input (PROM) during EPROM programming.	52		



# Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

## BLOCK DIAGRAM OF 8XC592 CAN CONTROLLER CIRCUITRY



# Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

## CAN FUNCTIONAL DESCRIPTION

### SIO1, CAN

#### (Controller Area Network)

CAN is the definition of a high performance communication protocol for serial data communication. The 8XC592 on-chip CAN Controller is a full implementation of the CAN-protocol. With the 8XC592, powerful local networks can be built, both for automotive and general industrial environments. This results in a strongly reduced wiring harness and enhanced diagnostic and supervisory capabilities.

### Features

- Multi-master architecture
- Bus access priority determined by the message identifier
- 2032 message identifier
- Guaranteed latency time for high priority messages
- Powerful error handling capability
- Data length from 0 up to 8 bytes
- Multicast and broadcast message facility
- Non-destructive bit-wise arbitration
- Non-return-to-zero (NRZ) coding/decoding with bit stuffing
- Programmable transfer rate (up to 1 Mbit/s)
- Programmable output driver configuration
- Suitable for use in a wide range of networks, including the SAE's network classes A, B and C
- DMA providing high-speed on-chip data exchange
- Bus failure management facility
- $AV_{DD/2}$  reference voltage

The CAN Controller meets the following automotive requirements:

- Short message length
- Guaranteed latency time\* for urgent messages
- Bus access priority, determined by the message identifier
- Powerful error handling capability
- Configuration flexibility to allow area network expansion.

### NOTE:

- \* The latency time defines the period between the initiation (Transmission Request) and the start of the transmission on the bus. The latency time strongly depends on a large variety of bus-related conditions. In the case of a message being transmitted on the bus and one distortion, the latency time can be up to 149 bit times (worst case). For more information, see the application note on bit timing.

### CAN Functional Overview

The 8XC592 includes all hardware modules necessary to implement the transfer layer which represents the kernel of the CAN protocol. Refer to the block diagram (previous page) of the CAN controller portion of the 8XC592.

### Interface Special Function Registers

The data transfer between the CPU and the CAN part of the 8XC592 is done via four SFRs:

CANADR (DBH):	to point to a register of the CAN-controller
CANDAT (DAH):	to read or write data
CANCON (D9H):	to read interrupt flags and to write commands
CANSTA (D8H):	to read status information and to write DMA pointer to the MAIN RAM

Additionally, the DMA-logic allows a high-speed data exchange between the CAN-controller and the MAIN RAM (see section "Handling of the CPU-CAN interface").

### Interface Management Logic (IML)

The IML interprets the commands from the CPU, controls the allocation of the message buffers Transmit Buffer (TBF), Receive Buffer 0 (RBF0), and Receive Buffer 1 (RBF1), and provides interrupts and status information to the CPU.

### Transmit Buffer (TBF)

The TBF is an interface between the CPU and the Bit Stream Processor (BSP) and is able to store a complete message. The buffer is written by the CPU and read by the BSP. The TBF is 10 bytes long to hold the Descriptor (2 bytes) and the Data-Field (up to 8 bytes) of the message.

### Receive Buffer (RBF0, RBF1)

The RBF is an interface between the BSP and the CPU and stores a message received from the busline. Once filled by BSP and allocated to the CPU by IML, the buffer cannot be used to store subsequently received messages until the CPU has (read and) released the buffer.

To reduce the requirements on the CPU, two receive buffers (RBF0, RBF1) are implemented. While one RBF is allocated to the CPU, the BSP may write to the other one. Both RBF0 and RBF1 are 10 bytes long to hold the Descriptor (2 bytes) and the Data-Field (up to 8 bytes) of the message.

### Bit Stream Processor (BSP)

This is a sequencer controlling the data stream between transmit and receive buffers (parallel data) and the busline (serial data). The BSP contains the Acceptance Filter and also controls the TCL and the EML such that the processes of reception, arbitration, transmission, and error signaling are performed according to the protocol. The BSP provides signals to the IML indicating when a message has got acceptance, when a receive buffer contains a valid message, and also when the transmit buffer is no longer required after a successful transmission.

### Bit Timing Logic (BTL)

This block monitors the busline using the (built-in) Input Comparator and handles the busline-related bit timing.

The BTL synchronizes on a "recessive" to "dominant" busline transition at the beginning of a message (hard synchronization) and resynchronizes on further transitions during the reception of a message (soft synchronization).

The BTL also provides programmable time segments to compensate for the propagation delay times and phase shifts (e.g., due to oscillator drifts) and to define the sampling time and the number of samples (one or three) to be taken within a bit time.

### Transceiver Logic (TCL)

The TCL controls the transmit output driver.

### Error Management Logic (EML)

The EML is responsible for the error confinement of the transfer-layer modules. The EML gets error announcements from BSP and then informs the BSP, TCL, and IML about error statistics.

## 80C592/83C592/87C592

The exchange of status, control and command signals between the CPU and the CAN Controller is performed in the control segment. The layout of this segment is shown in Figure 1. After an initial down-load, the contents of the registers Acceptance Code, Acceptance Mask, bus Timing 0 and 1 and Output Control should not be changed. These registers may only be accessed when the Reset Request bit in the Control Register is set HIGH (see "Control Register").

ADDRESS	CONTROL SEGMENT	DATA FIELD	TRANSMIT BUFFER	RECEIVE BUFFER 0 OR 1
0	CONTROL			
1	COMMAND			
2	STATUS			
3	INTERRUPT			
4	ACCEPTANCE CODE			
5	ACCEPTANCE MASK			
6	BUS TIMING 0			
7	BUS TIMING 1			
8	OUTPUT CONTROL			
9	TEST			
10	IDENTIFIER,			
11	RTR BIT, DATA LENGTH CODE			
12	BYTE 1			
13	BYTE 2			
14	BYTE 3			
15	BYTE 4			
16	BYTE 5			
17	BYTE 6			
18	BYTE 7			
19	BYTE 8			
20	IDENTIFIER,			
21	RTR BIT, DATA LENGTH CODE			
22	BYTE 1			
23	BYTE 2			
24	BYTE 3			
25	BYTE 4			
26	BYTE 5			
27	BYTE 6			
28	BYTE 7			
29	BYTE 8			

Figure 1. CAN Controller Internal Address Allocation

**Figure 1. CAN Controller Internal Address Allocation**

Single-chip 8-bit microcontroller  
with CAN controller

80C592/83C592/87C592

Table 1. CAN Registers

DESCRIPTION	ADDRESS	7 (MSB)	6	5	4	3	2	1	0 (LSB)
<b>Control Segment</b>									
Control Register	0	Test Mode	Synch	Reference Active	Overrun Interrupt Enable	Error Interrupt Enable	Transmit Interrupt Enable	Receive Interrupt Enable	Reset Request
Command Register	1	RX0 Active	RX1 Active	Wake-Up Mode	Sleep	Clear Overrun Status	Release Receive Buffer	Abort Trans-mission	Trans-mission Request
Status Register	2	Bus Status	Error Status	Transmit Status	Receive Status	Trans-mission Complete Status	Transmit Buffer Access	Data Overrun	Receive Buffer Status
Interrupt Register	3	reserved	reserved	reserved	Wake-Up Interrupt	Overrun Interrupt	Error Interrupt	Transmit Interrupt	Receive Interrupt
Acceptance Code Register	4	AC.7	AC.6	AC.5	AC.4	AC.3	AC.2	AC.1	AC.0
Acceptance Mask Register	5	AM.7	AM.6	AM.5	AM.4	AM.3	AM.2	AM.1	AM.0
Bus Timing Register 0	6	SJW.1	SJW.0	BRP.5	BRP.4	BRP.3	BRP.2	BRP.1	BRP.0
Bus Timing Register 1	7	SAM	TSEG2.2	TSEG2.1	TSEG2.0	TSEG1.3	TSEG1.2	TSEG1.1	TSEG1.0
Output Control Register	8	OCTP1	OCTN1	OCPOL1	OCTP0	COTN0	OCPOL0	OCMODE1	OCMODE0
<b>Transmit Buffer</b>									
Identifier	10	ID.10	ID.9	ID.8	ID.7	ID.6	ID.5	ID.4	ID.3
RTR, Data Length Code	11	ID.2	ID.1	ID.0	RTR	DLC.3	DLC.2	DLC.1	DLC.0
Bytes 1–8	12–19	Data	Data	Data	Data	Data	Data	Data	Data
<b>Receive Buffer 0/1</b>									
Identifier	20	ID.10	ID.9	ID.8	ID.7	ID.6	ID.5	ID.4	ID.3
RTR, Data Length Code	21	ID.2	ID.1	ID.0	RTR	DLC.3	DLC.2	DLC.1	DLC.0
Bytes 1–8	22–29	Data	Data	Data	Data	Data	Data	Data	Data

# Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

## Control Register (CR)

The contents of the Control Register are used to change the behavior of the CAN Controller. Control bits may be set or reset by the CPU which uses the Control Register as a read/write memory.

**Table 2. Description of the Control Register Bits**

CR	ADDRESS 0			
BIT	SYMBOL	NAME	VALUE	FUNCTION
CR.0	RR	Reset Request <sup>1</sup>	HIGH (present)	Detection of a Reset Request results in the CAN Controller aborting the current transmission/reception of a message entering the reset state.
			LOW (absent)	On the HIGH-to-LOW transition of the Reset Request bit, the CAN Controller returns to its normal operating state.
CR.1	RIE	Receive Interrupt Enable	HIGH (enabled)	When a message has been received without errors, the CAN Controller transmits a Receive Interrupt signal to the CPU.
			LOW (disabled)	No transmission of the Receive Interrupt signal by the CAN Controller to the CPU.
CR.2	TIE	Transmit Interrupt Enable	HIGH (enabled)	When a message has been successfully transmitted or the transmit buffer is accessible again, (e.g., after an Abort Transmission command) the CAN Controller transmits a Transmit Interrupt signal to the CPU.
			LOW (disabled)	No transmission of the Transmit Interrupt signal by the CAN Controller to the CPU.
CR.3	EIE	Error Interrupt Enable	HIGH (enabled)	If the Error or Bus Status change (see status Register), the CPU receives an Error Interrupt signal.
			LOW (disabled)	The CPU receives no Error Interrupt signal.
CR.4	OIE	Overrun Interrupt Enable	HIGH (enabled)	If the Data Overrun bit is set (see Status Register), the CPU receives an Overrun Interrupt signal.
			LOW (disabled)	The CPU receives no Overrun Interrupt signal from the CAN Controller.
CR.5	RA	Reference Active <sup>2</sup>	HIGH (output)	The pin REF is an AV <sub>DD/2</sub> reference output.
			LOW (input)	A reference voltage may be input.
CR.6	S	Synch <sup>2</sup>	HIGH (2 edges)	Bus-line transitions from recessive-to-dominant and vice versa are used for resynchronization.
			LOW (1 edge)	Only transitions from recessive-to-dominant are used for resynchronization.
CR.7	—	RESERVED		

### NOTES:

- During an external reset (RST = HIGH) or when the Bus Status bit is set HIGH (Bus-Off), the IML forces the Reset Request HIGH (present). During an external reset the CPU cannot set the Reset Request bit LOW (absent). Therefore, after having set the Reset Request bit LOW (absent), the CPU must check this bit to ensure that the external reset pin is not being held HIGH (present). After the Reset Request bit is set LOW (absent) the CAN controller will wait for:
  - one occurrence of the Bus-Free signal (11 recessive bits), if the preceding reset (Reset Request = HIGH) has been caused by an external reset or a CPU initiated reset.
  - 128 occurrences of Bus-Free, if the preceding reset (Reset Request = HIGH) has been caused by a CAN Controller initiated Bus-Off, before re-entering the Bus-On mode.
  - When Reset Request is set HIGH (present), for whatever reason, the control, command, status and interrupt bits are affected, see Table 3. Only, when Reset Request is set HIGH (present) the registers at addresses 4 to 8 are accessible.
- A modification of the bits Reference Active and Synch is only possible with Reset Request = HIGH (present). It is allowed to set these bits while Reset Request is changed from HIGH to LOW. After an external reset (pin RST = HIGH) the Reference Active bit is set HIGH (output), the Synch bit is undefined.



Single-chip 8-bit microcontroller  
with CAN controller

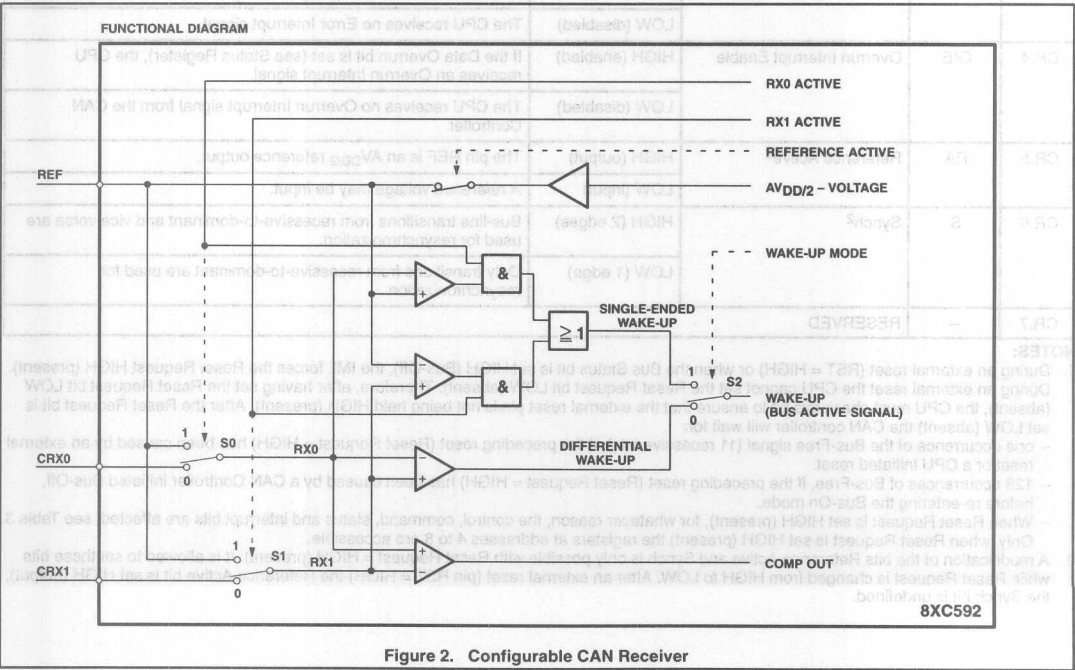
80C592/83C592/87C592

Table 3. Effects of Setting the Reset Request Bit HIGH (present)

TYPE	BIT	EFFECT
Control	Reference Active	HIGH (output), only after an external reset
Command	RX0 Active / RX1 Active	HIGH (RX0=CRX0, RX1=CRX1), only after an external reset
	Sleep	LOW (wake-up)
	Clear Overrun Status	HIGH (clear)
	Release Receive Buffer	HIGH (released)
	Abort Transmission	LOW (absent)
	Transmission Request	LOW (absent)
Status	Bus Status	LOW (Bus-On), only after an external reset
	Error Status	LOW (no error), only after an external reset
	Transmit Status	LOW (idle)
	Receive Status	LOW (idle)
	Transmission Complete Status	HIGH (complete)
	Transmit Buffer Access	HIGH (released)
Interrupt	Data Overrun	LOW (absent)
	Receive Buffer Status	LOW (empty)
	Overrun Interrupt	LOW (reset)
Interrupt	Transmit Interrupt	LOW (reset)
	Receive Interrupt	LOW (reset)

Command Register (CMR)

A command bit initiates an action within the transfer layer of the CAN controller, the Command Register appears to the CPU as a read/write memory, except of the bits CMR.0 to CMR.3, which return HIGH if being read.



# Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

Table 4. Description of the Command Register Bits

CMR		ADDRESS 1			
BIT	SYMBOL	NAME	VALUE		FUNCTION
CMR.0	TR	Transmission Request <sup>1</sup>	HIGH (present)		A message shall be transmitted.
			LOW (absent)		No action.
CMR.1	AT	Abort Transmission <sup>2</sup>	HIGH (present)		If not already in progress, a pending Transmission Request is cancelled.
			LOW (absent)		No action.
CMR.2	RRB	Release Receive Buffer <sup>3</sup>	HIGH (released)		The Receive Buffer attached to the CPU is released.
			LOW (no action)		No action.
CMR.3	COS	Clear Overrun <sup>4</sup>	HIGH (clear)		The Data Overrun status bit is set to LOW (see Status Register).
			LOW (no action)		No action.
CMR.4	SLP	Sleep <sup>5</sup>	HIGH (sleep)		The CAN Controller enters sleep mode if no CAN interrupt is pending and there is no bus activity.
			LOW (wake up)		The CAN Controller functions normally.
CMR.5	WUM	Wake-Up Mode <sup>6</sup>	HIGH (single ended)		The difference of the RX signals to the internal reference voltage AV <sub>DD/2</sub> is used for wake up.
			LOW (differential)		The differential signal between RX0 and RX1 is used for wake up.
CMR.6	RX1A	RX1 Active <sup>7</sup>	RX0 Active	RX1 Active	See Figure 2.
CMR.7	RX0A	RX0 Active <sup>7</sup>	1	1	RX0 = CRX0, RX1 = CRX1
			1	0	RX0 = CRX0, RX1 = AV <sub>DD/2</sub>
			0	1	RX0 = AV <sub>DD/2</sub> , RX1 = CRX1
			0	0	No action.

## NOTES:

1. If the Transmission Request bit was set HIGH in a previous command, it cannot be cancelled by setting the Transmission Request bit LOW (absent). Cancellation of the requested transmission may be performed by setting the Abort Transmission bit HIGH (present).
2. The Abort Transmission bit is used when the CPU requires the suspension of the previously requested transmission, e.g., to transmit an urgent message. A transmission already in progress is not stopped. In order to see if the original message had been either transmitted successfully or aborted, the Transmission Complete Status bit should be checked. This should be done after the Transmit Buffer Access bit has been set HIGH (released) or a Transmit Interrupt has been generated (see Interrupt Register).
3. After reading the contents of the Receive Buffer (RBF0 or RBF1) the CPU must release this buffer by setting Release Receive Buffer bit HIGH (released). This may result in another message becoming immediately available.
4. This command bit is used to acknowledge the Data Overrun condition signaled by the Data Overrun status bit. It may be given or set at the same time as a Release Receive Buffer command bit.
5. The CAN Controller will enter sleep mode, if the Sleep bit is set HIGH (sleep), there is no bus activity and no interrupt is pending. A CAN Controller will wake up after the Sleep bit is set LOW (wake up) or when there is bus activity. On wake up, a Wake-Up Interrupt (see Interrupt Register) is generated (see "Power Reduction Modes"). A CAN Controller which is sleeping and then awoken by bus activity will not be able to receive this message until it detects a Bus-Free signal. The Sleep bit, if being read, reflects the status of the CAN Controller.
6. The Wake-Up Mode bit should be set at the same time as the Sleep bit. The differential wake up mode is useful if both bus wires are fully functioning; it minimizes the probability of wake ups due to noise. The single ended wake up mode is recommended if a wake up must be possible even if one bus wire is already or may become disturbed (See Figure 2).
7. The RX0/RX1 Active bits, if being read, reflect the status of the respective switches (See Figure 2). It is recommended to change the switches only during the reset state (Reset Request is HIGH).

# Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

## Status Register (SR)

The contents of the Status Register reflects the status of the CAN Controller. The Status Register appears to the CPU as a read only memory.

**Table 5. Description of the Status Register Bits**

SR	ADDRESS 2			
BIT	SYMBOL	NAME	VALUE	FUNCTION
SR.0	RBS	Receive Buffer Status <sup>1</sup>	HIGH (full)	This bit is set when a new message is available.
			LOW (empty)	No message has become available since the last Release Receive Buffer command bit was set.
SR.1	DO	Data Overrun <sup>2</sup>	HIGH (overrun)	This bit is set HIGH (overrun) when both Receive Buffers are full and the first byte of another message should be stored.
			LOW (absent)	No data overrun has occurred since the Clear Overrun command was given.
SR.2	TBS	Transmit Buffer Access <sup>3</sup>	HIGH (released)	The CPU may write a message into the TBF.
			LOW (locked)	The CPU cannot access the Transmit Buffer. A message is either waiting for transmission or is in the process of being transmitted.
SR.3	TCS	Transmit Complete Status <sup>3</sup>	HIGH (complete)	Last requested transmission has been successfully completed.
			LOW (incomplete)	Previously requested transmission is not yet completed.
SR.4	RS	Receive Status <sup>4</sup>	HIGH (receive)	The CAN Controller is receiving a message.
			LOW (idle)	No message is received.
SR.5	TS	Transmit Status <sup>4</sup>	HIGH (transmit)	The CAN Controller is transmitting a message.
			LOW (idle)	No message is transmitted.
SR.6	ES	Error Status	HIGH (error)	At least one of the Error Counters has reached the CPU Warning limit.
			LOW (ok)	Both Error Counters have not reached the warning limit.
SR.7	BS	Bus Status <sup>5</sup>	HIGH (Bus-Off)	The CAN Controller is not involved in bus activities.
			LOW (Bus-On)	The CAN Controller is involved in bus activities.

### NOTES:

- If the command bit Release Receive Buffer is set HIGH (released) by the CPU, the Receive Buffer Status bit is set LOW (empty) by IML. When a new message is stored in any of the receive buffers, the Receive Buffer Status bit is set HIGH (full) again.
- If Data Overrun = HIGH (overrun) is detected, the currently received message is dropped. A transmitted message, granted acceptance, is also stored in a Receive Buffer. This occurs because it is not known if the CAN Controller will lose arbitration and so become a receiver of the message. If no Receive Buffer is available, Data Overrun is signaled. However, this transmitted and accepted message does neither cause a Receive Interrupt nor set the Receive Buffer Status bit to HIGH (full). Also, a Data Overrun does not cause the transmission of an Overload Frame.
- If the CPU tries to write to the Transmit Buffer when the Transmit Buffer Access bit is LOW (locked), the written bytes will not be accepted and will be lost without this being signaled. The Transmission Complete Status bit is set LOW (incomplete) whenever the Transmission Request bit is set HIGH (present). If an Abort Transmission command is issued, the Transmit Buffer will be released. If the message, which was requested and then aborted, was not transmitted, the Transmission Complete Status bit will remain LOW.
- If both the Receive Status and Transmit Status bits are LOW (idle) the CAN-bus is idle.
- When the Bus Status bit is set HIGH (Bus-Off), the CAN Controller will set the Reset Request bit HIGH (present). It will stay in this state until the CPU sets the Reset Request bit LOW (absent). Once this is completed, the CAN Controller will wait the minimum protocol-defined time (128 occurrences of the Bus-Free signal) before setting the Bus Status bit LOW (Bus-On), the Error Status bit LOW (ok), and resetting the Error Counters.

# Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

## Interrupt Register (IR)

The Interrupt Register allows the identification of an interrupt source. When one or more bits of this register are set, a CAN interrupt (SIO1) will be indicated to the CPU. All bits are reset by the CAN Controller after this register is read by the CPU. This register appears to the CPU as a read only memory.

**Table 6. Description of the Interrupt Register Bits**

IR	ADDRESS 3			
BIT	SYMBOL	NAME	VALUE	FUNCTION
IR.0	RI	Receive Interrupt <sup>1</sup>	HIGH (set)	This bit is set when a new message is available in the Receive Buffer and the Receive Interrupt Enable bit is HIGH (enabled).
			LOW (reset)	Receive Interrupt bit is automatically reset by a read access of Interrupt Register by the CPU.
IR.1	TI	Transmit Interrupt	HIGH (set)	This bit is set on a change of the Transmit Buffer Access from LOW to HIGH (released) and Transmit Interrupt Enable is HIGH (enabled).
			LOW (reset)	Transmit Interrupt bit will be reset after a read access of the Interrupt Register by the CPU.
IR.2	EI	Error Interrupt	HIGH (set)	This bit is set on a change of either the Error Status or Bus Status bits (see Status Register) if the Error Interrupt Enable is HIGH (enabled).
			LOW (reset)	The Error Interrupt bit is reset by a read access of the Interrupt Register by the CPU.
IR.3	OI	Overrun Interrupt <sup>2</sup>	HIGH (set)	This bit is set HIGH, if both Receive Buffers contain a message and the first byte of another message should be stored (passed acceptance), and the Overrun Interrupt Enable is HIGH (enabled).
			LOW (reset)	Overrun Interrupt bit is reset by a read access of the Interrupt register by the CPU.
IR.4	WUI	Wake-Up Interrupt	HIGH (set)	The Wake-Up Interrupt bit is set HIGH when the sleep mode is left (see Command Register).
			LOW (reset)	Wake-Up Interrupt bit is reset by a read access of the Interrupt Register by the CPU.
IR.5	—	RESERVED		
IR.6	—	RESERVED		
IR.7	—	RESERVED		

### NOTES:

1. Receive Interrupt bit (if enabled) and Receive Buffer Status bit (see Status Register) are set at the same time.
2. Overrun Interrupt bit (if enabled) and Data Overrun bit (see Command Register) are set at the same time.



## Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

### Acceptance Code Register (ACR)

The Acceptance Code Register is part of the acceptance filter of the CAN Controller. This register can be accessed (read/write), if the Reset Request bit is set HIGH (present). When a message is received which passes the acceptance test and if there is an empty Receive Buffer, then the respective Descriptor and Data Field (see Figure 1) are sequentially stored in this empty buffer. In the case that there is no empty Receive buffer, the Data Overrun bit is set HIGH (overrun), see Status and Interrupt Register. When the complete message has been correctly received, the following occurs:

- The Receive Buffer Status bit is set HIGH (full)
- If the Receive Interrupt Enable bit is set HIGH (enabled), the receive Interrupt is set HIGH (set).

The Acceptance Code bits (AC.7–AC.0) and the eight most significant bits of the message's Identifier (ID.10–ID.3) must be equal to those bit positions which are marked relevant by the Acceptance Mask bits (AM.7–AM.0). If the following equation is satisfied, acceptance is given:

$$[(ID.10 \dots ID.3) \text{ EQUAL } (AC.7 \dots AC.0)] \text{ or } (AM.7 \dots AM.0) = 1111 \ 1111_b$$

#### Acceptance Code Register Bits

ACR	ADDRESS 4							
	7	6	5	4	3	2	1	0
	AC.7	AC.6	AC.5	AC.4	AC.3	AC.2	AC.1	AC.0

During transmission of a message which passes the acceptance test, the message is also written to its own Receive Buffer. If no Receive Buffer is available, Data Overrun is signaled because it is not known at the start of a message whether the CAN Controller will lose arbitration and so become a receiver of the message.

### Acceptance Mask Register (AMR)

The Acceptance Mask Register is part of the acceptance filter of the CAN Controller. This register can be accessed (read/write) if the Reset Request bit is set HIGH (present). The

Acceptance Mask Register qualifies which of the corresponding bits of the acceptance code are "relevant" or "don't care" for acceptance filtering.

#### Acceptance Mask Register Bits

AMR	ADDRESS 5							
	7	6	5	4	3	2	1	0
	AM.7	AM.6	AM.5	AM.4	AM.3	AM.2	AM.1	AM.0

#### Description of the Acceptance Mask Register Bits

ACCEPTANCE MASK BIT	VALUE	COMMENTS
AM.7 to AM.0	HIGH (don't care)	This bit position is "don't care" for the acceptance of a message.
	LOW (relevant)	This bit position is "relevant" for acceptance filtering.

### Bus Timing Register 0 (BTR 0)

The contents of Bus Timing Register 0 defines the values of the Baud Rate Prescaler (BRP) and the Synchronization Jump Width (SJW). This register can be accessed (read/write) if the Reset Request bit is set HIGH (present).

#### Bus Timing Register 0 Bits

BTR0	ADDRESS 6							
	7	6	5	4	3	2	1	0
	SJW.1	SJW.0	BRP.5	BRP.4	BRP.3	BRP.2	BRP.1	BRP.0

#### Baud Rate Prescaler (BRP)

The period of the system clock  $t_{SCL}$  is programmable and determines the individual bit timing. The system clock is calculated using the following equation:

$$t_{SCL} = 2t_{CLK} (32BRP.5 + 16BRP.4 + 8BRP.3 + 4BRP.2 + 2BRP.1 + BRP.0 + 1)$$

$t_{CLK}$  = time period of the 8XC592 oscillator.

#### Synchronization Jump Width (SJW)

To compensate for phase shifts between clock oscillators of different bus controllers, any bus controller must resynchronize on any relevant signal edge of the current

transmission. The synchronization jump width defines the maximum number of clock cycles a bit period may be shortened or lengthened by one resynchronization:

$$t_{SJW} = t_{SCL} (2SJW.1 + SJW.0 + 1)$$

### Bus Timing Register 1 (BTR1)

The contents of Bus Timing Register 1 defines the length of the bit period, the location of the sample point and the number of samples to be taken at each sample point. This register can be accessed (read/write) if the Reset Request bit is set HIGH (present).

#### Bus Timing Register 1 Bits

BTR1	ADDRESS 7			
	7	6	5	4
	SAM	TSEG2.2	TSEG2.1	TSEG2.0
		3	2	1
		TSEG1.3	TSEG1.2	TSEG1.1
				0
				TSEG1.0

#### Sampling (SAM)

BIT	VALUE	COMMENTS
SAM	HIGH (3 samples)	3 samples are taken.
	LOW (1 sample)	The bus is sampled once.

SAM = LOW is recommended for high speed buses (SAE class C), while SAM = HIGH is recommended for slow/medium speed buses (class A and B) where filtering of spikes on the bus-line is beneficial (see "Bit Timing Restrictions").

#### Time Segment 1 (TSEG1) and Time Segment 2 (TSEG2)

TSEG1 and TSEG2 determine the number of clock cycles per bit period and the location of the sample point:

$$t_{TSEG1} = t_{SCL} (8TSEG1.3 + 4TSEG1.2 + 2TSEG1.1 + TSEG1.0 + 1)$$

$$t_{TSEG2} = t_{SCL} (4TSEG2.2 + 2TSEG2.1 + TSEG2.0 + 1)$$

For further information on bus timing see Bus Timing Register 0 and "Bus Timing/Synchronization".



## Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

### Output Control Register (OCR)

The Output Control Register allows, under software control, the set-up of different output driver configurations. This register can be accessed (read/write) if the Reset Request bit is set HIGH (present).

#### Output Control Register Bits

OCR	ADDRESS 8				
7	6	5	4	3	
OCTP1	OCTN1	OCPOL1	OCTP0	OCTN0	
	2	1		0	
	OCPOL0	OCMODE1	OCMODE0		

If the CAN Controller is in the sleep mode (Sleep = HIGH) a recessive level is output on the CTX0 and CTX1 pins. If the CAN Controller is in the reset state (Reset Request = HIGH) the output drivers are floating.

#### Normal Output Mode

In Normal Output Mode the bit sequence (TXD) is sent via CTX0 and CTX1. The

voltage levels on the output driver pins CTX0 and CTX1 depend on both the driver characteristic programmed by OCTPx, OCTNx (float, pull-up, pull-down, push-pull) and the output polarity programmed by OCPOLx (see Figure 3).

#### Clock Output Mode

For the CTX0 pin this is the same as in Normal Output Mode. However, the data stream to CTX1 is replaced by the transmit clock (TXCLK). The rising edge of the transmit clock (non-inverted) marks the beginning of a bit period. The clock pulse width is  $t_{\text{scL}}$ .

#### Bi-phase Output Mode

In contrast to Normal Output Mode the bit representation is time variant and toggled. If the bus controllers are galvanically decoupled from the bus-line by a transformer, the bit stream is not allowed to contain a DC component. This is achieved by the following

scheme. During recessive bits all outputs are de-activated (floating). Dominant bits are sent alternatingly on CTX0 and CTX1, i.e., the first dominant bit is sent on CTX0, the second is sent on CTX1, and the third one is sent on CTX0 again, etc.

#### Test Output Mode

For the CTX0 pin this is the same as in Normal Output Mode. To measure the delay time of the transmitter and receiver this mode connects the output of the input comparator (COMP OUT) with the input of the output driver CTX1. This mode is used for production testing only.

The following two tables, Table 7 and Table 8, show the relationship between the bits of the Output Control Register and the two serial output pins CTX0 and CTX1 of the 8XC592 – CAN Controller, connected to the serial bus (see Block Diagram, page 855).

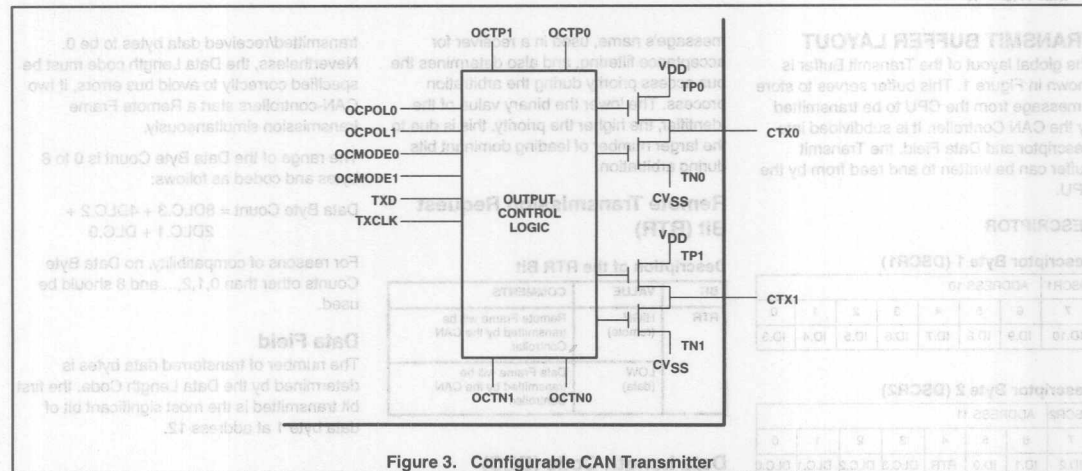


Figure 3. Configurable CAN Transmitter

Table 7. Description of the Output Mode Bits

OCMODE1	OCMODE0	DESCRIPTION
1	0	Normal output Mode; CTX0, CTX1: bit sequence (TXD; note 1)
1	1	Clock output Mode; CTX0: bit sequence, CTX1: bus clock (TXCLK)
0	0	Bi-phase output Mode
0	1	Test output Mode; CTX0: bit sequence, CTX1: COMP OUT

#### NOTE:

1. TXD is the data bit to be transmitted.

# Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

Table 8. Output Pin Set-Up

DRIVE	OCTPx	OCTNx	OCPOLx	TXD	TPx	TNx	CTXx
Float	0	0	0	0	Off	Off	float
	0	0	0	1	Off	Off	float
	0	0	1	0	Off	Off	float
	0	0	1	1	Off	Off	float
Pull-down	0	1	0	0	Off	On	LOW
	0	1	0	1	Off	Off	float
	0	1	1	0	Off	Off	float
	0	1	1	1	Off	On	LOW
Pull-up	1	0	0	0	Off	Off	float
	1	0	0	1	On	Off	HIGH
	1	0	1	0	On	Off	HIGH
	1	0	1	1	Off	Off	float
Push/Pull	1	1	0	0	Off	On	LOW
	1	1	0	1	On	Off	HIGH
	1	1	1	0	On	Off	HIGH
	1	1	1	1	Off	On	LOW

## NOTES:

1. TPx is the on-chip output transistor x, connected to  $V_{DD}$ ; x = 0 or 1.
2. TNx is the on-chip output transistor x, connected to  $CV_{SS}$ ; x = 0 or 1.
3. CTXx is the serial output level on CTX0 or CTX1. It is required that the output level on the CAN bus is dominant with TXD = 0 and recessive with TXD = 1.

## TRANSMIT BUFFER LAYOUT

The global layout of the Transmit Buffer is shown in Figure 1. This buffer serves to store a message from the CPU to be transmitted by the CAN Controller. It is subdivided into Descriptor and Data Field. The Transmit Buffer can be written to and read from by the CPU.

### DESCRIPTOR

#### Descriptor Byte 1 (DSCR1)

DSCR1	ADDRESS 10						
7	6	5	4	3	2	1	0
ID.10	ID.9	ID.8	ID.7	ID.6	ID.5	ID.4	ID.3

#### Descriptor Byte 2 (DSCR2)

DSCR2	ADDRESS 11						
7	6	5	4	3	2	1	0
ID.2	ID.1	ID.0	RTR	DLC.3	DLC.2	DLC.1	DLC.0

### Identifier (ID)

The Identifier consists of 11 bits (ID.10 to ID.0). ID.10 is the most significant bit, which is transmitted first on the bus during the arbitration process. The Identifier acts as the

message's name, used in a receiver for acceptance filtering, and also determines the bus access priority during the arbitration process. The lower the binary value of the Identifier, the higher the priority. This is due to the larger number of leading dominant bits during arbitration.

### Remote Transmission Request Bit (RTR)

#### Description of the RTR Bit

BIT	VALUE	COMMENTS
RTR	HIGH (remote)	Remote Frame will be transmitted by the CAN Controller.
	LOW (data)	Data Frame will be transmitted by the CAN Controller.

### Data Length Code (DLC)

The number of bytes (Data Byte Count) in the Data Field of a message is coded by the Data Length Code. At the start of a Remote Frame transmission, the Data Length Code is not considered due to the RTR bit being HIGH (remote). This forces the number of

transmitted/received data bytes to be 0. Nevertheless, the Data Length code must be specified correctly to avoid bus errors, if two CAN-controllers start a Remote Frame transmission simultaneously.

The range of the Data Byte Count is 0 to 8 bytes and coded as follows:

$$\text{Data Byte Count} = 8\text{DLC.3} + 4\text{DLC.2} + 2\text{DLC.1} + \text{DLC.0}$$

For reasons of compatibility, no Data Byte Counts other than 0, 1, 2, ..., and 8 should be used.

### Data Field

The number of transferred data bytes is determined by the Data Length Code. The first bit transmitted is the most significant bit of data byte 1 at address 12.

## RECEIVE BUFFER LAYOUT

The layout of the Receive Buffer and the individual bytes correspond to the definitions given for the Transmit Buffer layout, except that the addresses start at 20 instead of 10 (see Figure 1).

# Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

**Table 9. The Special Function Registers Between CPU and CAN**

SFR	ADR	ACS	MSB							LSB	
			7	6	5	4	3	2	1	0	
CANADR	DBh	RW	DMA	reserved <sup>3</sup>	Auto Inc	CANA4	CANA3	CANA2	CANA1	CANA0	
CANDAT	DAh	RW	CAND7	CAND6	CAND5	CAND4	CAND3	CAND2	CAND1	CAND0	
CANCON <sup>1</sup>	D9h	R	reserved <sup>3</sup>	reserved <sup>3</sup>	reserved <sup>3</sup>	Wake Up Interrupt	Overrun Interrupt	Error Interrupt	Transmit Interrupt	Receive Interrupt	
		W	RX0 Active	RX1 Active	Wake Up Mode	Sleep	Clear Overrun	Release RxBuf	Abort Transm	Transm Request	
CANSTA <sup>1,2</sup>	D8h	R	Bus Status	Error Status	Transmit Status	Receive Status	TxComp1 Status	TxBuf Access	Data Overrun	RxBuf Status	
		W	RAMA7	RAMA6	RAMA5	RAMA4	RAMA3	RAMA2	RAMA1	RAMA0	

**NOTES:**

1. Do not use a RMW instruction.
2. The bit addresses of CANSTA (7:0) are DFH...D8H.
3. Reserved bits are read as HIGH.

## HANDLING OF THE CPU-CAN INTERFACE

Via the four special registers CANADR, CANDAT, CANCON and CANSTA, the CPU has access to the CAN Controller and also to the DAM-logic. Note that CANCON and CANSTA have different meanings for a read and write access.

**CANADR**

The five least significant bits CANADR.4 down to CANADR.0 (CANA4...CANA0) define the address of one of the CAN Controller internal registers to be accessed via CANDAT. For instance, after an external hardware (e.g., power-on) reset CANADR contains the value 64h, and hence the CPU accesses (read/write) the Acceptance Code register of the CAN Controller, via the Special Function Register CANDAT. CANADR also controls the auto address increment mode via bit CANADR.5 (AutoInc) and the DMA-logic via bit CANADR.7 (DMA). CANADR is implemented as a read/write register.

**CANDAT**

The Special Function Register CANDAT appears as a port to the CAN Controller's internal register (memory location) being selected by CANADR. Reading or writing CANDAT is effectively an access to that CAN Controller internal register, which is selected by CANADR. CANDAT is implemented as a read/write register.

**CANCON**

When reading CANCON the Interrupt Register of the CAN Controller is accessed, while writing to CANCON means an access

to the Command Register. CANCON is implemented as a read/write register.

**CANSTA**

Reading CANSTA is an access to the Status Register of the CAN Controller. Writing to CANSTA sets the address of the on-chip Main RAM (internal data memory) for a subsequent DMA transfer. CANSTA is implemented as a bit-addressable read/write register.

**Auto Address Increment**

With the auto address increment mode a fast stack-like reading and writing of CAN Controller internal registers is provided. If the bit CANADR.5 (AutoInc) is high, the content of CANADR is incremented automatically after any read or write access to CANDAT. For instance, loading a message into the Transmit Buffer can be done by writing 2Ah into CANADR and then moving byte by byte of the message to CANDAT.

Incrementing CANADR beyond xx111111<sub>b</sub> resets the bit CANADR.5 (AutoInc) automatically (CANADR = xx000000<sub>b</sub>).

**High Speed DMA**

The DMA-logic allows to transfer a complete message (up to 10 bytes) between CAN Controller and Main RAM in 2 instruction cycles at maximum; up to 4 bytes are transferred in 1 instruction cycle. The performance of the CPU is strongly enhanced because this very fast transfer is done in the background. A DMA transfer is achieved by first writing the RAM address (0...FFh) into CANSTA and then setting the TX- or RX-Buffer address in CANADR and

the bit CANADR.7 (DMA) simultaneously; the RAM address points to the location of the first byte to be transferred. Setting the DMA bit causes an automatic evaluation of the Data Length Code and then the transfer; for a TX-DMA transfer the Data Length Code is expected at the location "RAM address + 1".

In order to program a TX-DMA transfer, the value 8Ah (address 10) has to be written into CANADR. Then a complete message, consisting of the 2-byte Descriptor and the Data Field (0...8 bytes), starting at location "RAM address" is transferred to the TX-Buffer.

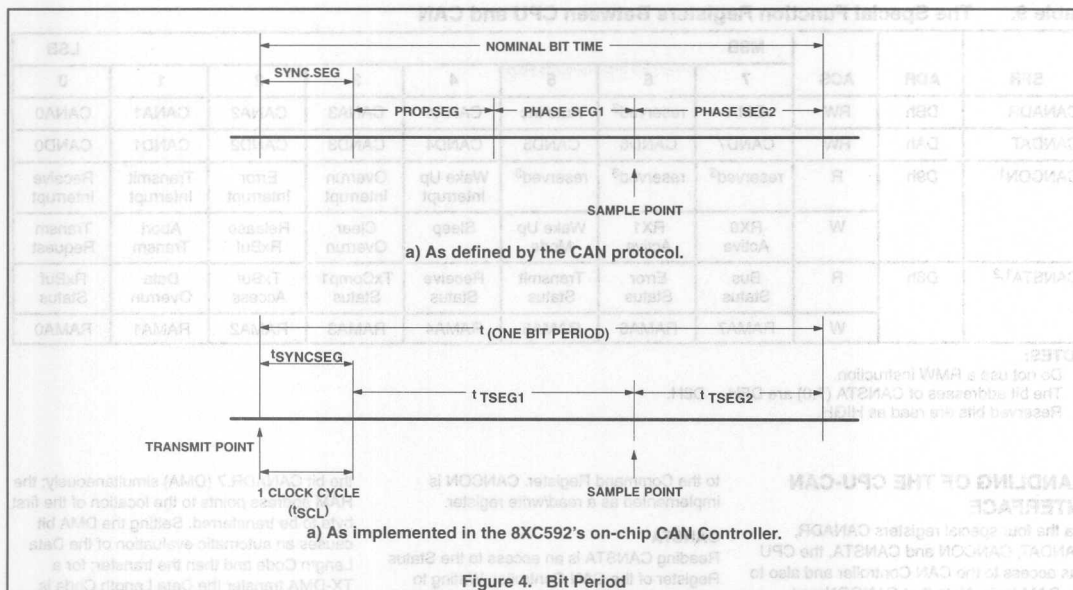
The RX-DMA transfer is very versatile. By writing a value in the range of 94H (address 20) up to 9DH (address 29) of CANADR the whole or a part of the received message, starting at the specified address, is transferred to the internal data memory. This allows e.g. to transfer the bytes of the Data Field only.

After a successful DMA transfer the DMA- (and Auto Inc-) bit is reset.

During a DMA transfer the CPU can process the next instruction. However, an access to the data memory, CANADR, CANDAT, CANCON or CANSTA is not allowed. After having set the DMA-bit, every interrupt is disabled until the end of the transfer. Note, that disadvantageous programming may lead to an interrupt response time of at most 10 instruction cycles. The shortest interrupt response time is achieved by using a 2-cycle instructions directly after setting the DMA-bit. During the reset state (bit Reset Request is HIGH) a DMA transfer is not possible.

# Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592



## BUS TIMING / SYNCHRONIZATION

The Bus Timing Logic (BTL) monitors the serial bus-line via the on-chip input comparator and performs the following functions:

- Monitors the serial bus-line level
- Adjusts the sample point, within a bit period (programmable)
- Samples the bus-line level using majority logic (programmable, 1 or 3 samples)
- Synchronization to the bit stream:
  - Hard synchronization at the start of a message
  - Resynchronization during transfer of a message

The configuration of the BTL is performed during the initialization of the CAN Controller. The BTL uses the following three registers:

- Control register (Synch)
- Bus Timing Register 0
- Bus Timing Register 1

## Bit Timing

A bit period is built up from a number of system clock cycles ( $t_{SCL}$ ) (see "Bus Timing Register 0"). One bit period is the result of the addition of the programmable segments TSEG1 and TSEG2 and the general segment SYNCSEG.

## Synchronization Segment (SYNCSEG)

The incoming edge of a bit is expected during this state; this state corresponds to one system clock cycle ( $1 \times t_{SCL}$ ).

## Time Segment 1 (TSEG1)

This segment determines the location of the sampling point within a bit period, which is at the end of TSEG1. TSEG1 is programmable from 1 to 16 system clock cycles.

The correct location of the sample point is essential for the correct functioning of a transmission. The following points must be taken into consideration:

- A Start-Of-Frame causes all CAN Controllers to perform a "hard synchronization" on the first recessive-to-dominant edge. During arbitration, however, several CAN Controllers may simultaneously transmit. Therefore it may require twice the sum of bus-line, input comparator and the output driver delay times until the bus is stable. This is the propagation delay time.
- To avoid sampling at an incorrect position, it is necessary to include an additional synchronization buffer on both sides of the sample point. The main reasons for incorrect sampling are:
  - Incorrect synchronization due to spikes on the bus-line;

- Slight variations in the oscillator frequency of each CAN Controller in the network, which results in a phase error.

- Time Segment 1 consists of the segment for compensation of propagation delays and the synchronization buffer segment directly before the sample point (see Figure 4).

## Time Segment 2 (TSEG2)

This time segment provides:

- Additional time at the sample point for calculation of the subsequent bit levels (e.g., arbitration);
- Synchronization buffer segment directly after the sample point.

TSEG2 is programmable from 1 to 8 system clock cycles.

## Synchronization Jump Width (SJW)

SJW defines the maximum number of clock cycles ( $t_{SCL}$ ) a period may be reduced or increased by one resynchronization. SJW is programmable from 1 to 4 system clock cycles.



## Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

### Propagation Delay Time ( $t_{PROP}$ )

The propagation delay time is calculated by summing the maximum propagation delay times of the physical bus, the input comparator and the output driver. The resulting sum is multiplied by 2 and then rounded up to the nearest multiple of  $t_{SCL}$ .

$$t_{PROP} = 2 \times (\text{physical bus delay} \\ + \text{input comparator delay} \\ + \text{output driver delay})$$

### Bit Timing Restrictions

Restrictions on the configuration of the bit timing are based on internal processing. The restrictions are:

- $t_{TSEG2} \geq 2t_{SCL}$
- $t_{TSEG2} \geq t_{SJW}$
- $t_{TSEG1} \geq t_{TSEG2}$
- $t_{TSEG1} \geq t_{SJW} + t_{PROP}$

The three sample mode ( $SAM = 1$ ) has the effect of introducing a delay of one system clock cycle on the bus-line. This must be taken into account for the correct calculation of  $TSEG1$  and  $TSEG2$ :

- $t_{TSEG1} \geq t_{SJW} + t_{PROP} + 2t_{SCL}$
- $t_{TSEG2} \geq 3t_{SCL}$

### Synchronization

Synchronization is performed by a state machine which compares the incoming edge with its actual bit timing and adapts the bit timing by hard synchronization or resynchronization.

#### Hard Synchronization

This type of synchronization occurs only at the beginning of a message. The CAN

Controller synchronizes on the first incoming recessive-to-dominant edge of a message (being the leading edge of a message's Start-Of-Frame bit).

### Resynchronization

Resynchronization occurs during the transmission of a message's bit stream to compensate for:

- Variations in individual CAN Controller oscillator frequencies;
- Changes introduced by switching from one transmitter to another (e.g., during arbitration).

As a result of resynchronization, either  $t_{TSEG1}$  may be increased by up to a maximum of  $t_{SJW}$ , or  $t_{TSEG2}$  may be decreased by up to a maximum of  $t_{SJW}$ :

- $t_{TSEG1} \leq t_{SCL} [(TSEG1 + 1) + (SJW + 1)]$
- $t_{TSEG2} \geq t_{SCL} [(TSEG2 + 1) - (SJW + 1)]$

**NOTE:**  $TSEG1$ ,  $TSEG2$  and  $SJW$  are the programmed numerical values.

The phase error ( $e$ ) of an edge is given by the position of the edge relative to  $SYNCSEG$ , measured in system clock cycles ( $t_{SCL}$ ). The value of the phase error is defined as:

- $e = 0$ , if the edge occurs within  $SYNCSEG$
- $e > 0$ , if the edge occurs within  $TSEG1$
- $e < 0$ , if the edge occurs within  $TSEG2$ .

The effect of resynchronization is:

- The same as that of a hard synchronization, if the magnitude of the

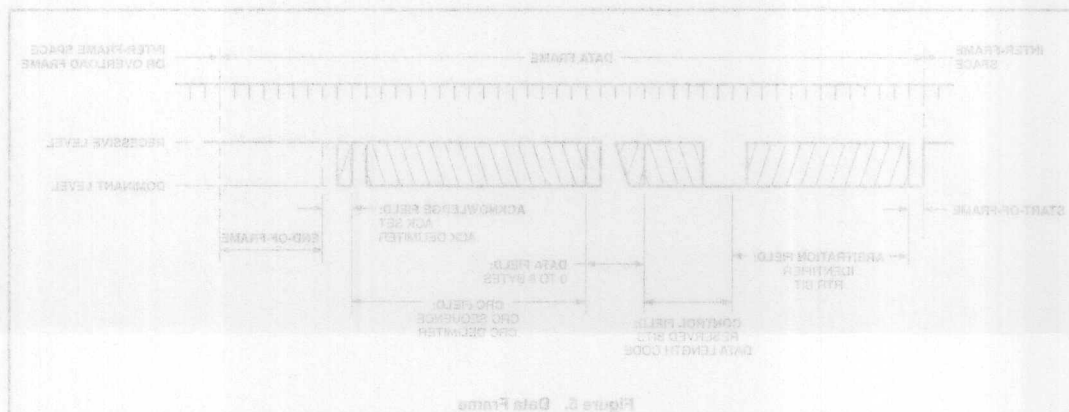
phase error ( $e$ ) is less or equal to the programmed value of  $t_{SJW}$ ;

- To increase a bit period by the amount of  $t_{SJW}$ , if the phase error is positive and the magnitude of the phase error is larger than  $t_{SJW}$ ;
- To decrease a bit period by the amount of  $t_{SJW}$ , if the phase error is negative and the magnitude of the phase error is larger than  $t_{SJW}$ .

### Synchronization Rules

The synchronization rules are as follows:

- Only one synchronization within one bit time used.
- An edge is used for synchronization only if the value detected at the previous sample point differs from the bus value immediately after the edge.
- Hard synchronization is performed whenever there is a recessive-to-dominant edge during Bus-Idle.
- All other edges (recessive-to-dominant and optionally dominant-to-recessive edges if the Synch bit is set HIGH) which are candidates for resynchronization will be used with the following exception:
  - A transmitting CAN Controller will not perform a resynchronization as a result of a recessive-to-dominant edge with positive phase error, if only these edges are used for resynchronization. This ensures that the delay times of the output driver and input comparator do not cause a permanent increase in the bit time.





## Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

### CAN-PROTOCOL DESCRIPTION

#### Frame Types

The 8XC592's CAN Controller supports the four different CAN protocol frame types for communication:

- Data Frame, to transfer data
- Remote Frame, request for data
- Error Frame, globally signals a (locally) detected error condition
- Overload Frame, to extend delay time of subsequent frames (an Overload Frame is not initiated by the 8XC592 CAN Controller).

#### Bit Representation

There are two logical bit representations used in the CAN protocol:

- A recessive bit on the bus-line appears only if all connected CAN Controllers send a recessive bit at that moment
- Dominant bits always overwrite recessive bits, i.e., the resulting bit level on the bus-line is dominant.

#### Data Frame

A Data Frame carries data from a transmitting CAN Controller to one or more receiving ones. A Data Frame is composed of seven different bit-fields:

- Start-Of-Frame
- Arbitration Field
- Control Field
- Data Field (may have a length of zero)
- CRC Field
- Acknowledge Field
- End-Of-Frame

#### Start-of-Frame Bit

Signals the start of a Data Frame or Remote Frame. It consists of a single dominant bit used for hard synchronization of a CAN Controller in receive mode.

#### Arbitration Field

Consists of the message Identifier and the RTR bit. In the case of simultaneous message transmissions by two or more CAN Controllers the bus access conflict is solved by bit-wise arbitration, which is active during the transmission of the Arbitration Field.

#### Identifier

This 11-bit field is used to provide information about the message, as well as the bus access priority. It is transmitted in the order ID.10 to ID.0 (LSB). The situation that the seven most significant bits (ID.10 to ID.4) are all recessive must not occur.

An Identifier does not define which particular CAN Controller will receive the frame because a CAN-based communication network does not differentiate between a point-to-point, multicast or broadcast communication.

#### RTR Bit

A CAN Controller, acting as a receiver for certain information may initiate the transmission of the respective data by transmitting a Remote Frame to the network, addressing the data source via the Identifier and setting the RTR bit HIGH (remote; recessive bus level). If the data source simultaneously transmits a Data Frame containing the requested data, it uses the same Identifier. No bus access conflict occurs due to the RTR bit being set LOW (data; dominant bus level) in the Data Frame.

#### Control Field

This field consists of six bits. It includes two reserved bits (for future expansions of the CAN protocol), transmitted with a dominant bus level, and is followed by the Data Length Code (4 bits). The number of bytes (destuffed; number of data bytes to be transmitted/received) in the Data Field is indicated by the Data Length Code. Admissible values of the Data Length Code, and hence the number of bytes in the (destuffed) Data Field, are (0, 1, ..., 8). A logic 0 (logic 1) in the Data Length Code is transmitted as dominant (recessive) bus level, respectively.

#### Data Field

The data, stored within the Data Field of the transmit buffer, are transmitted according to the Data Length Code. Conversely, data of a received Data Frame will be stored in the Data Field of a Receive Buffer. The Data Field can contain from 0 up to 8 bytes. The most significant bit of the first data byte (lowest address) is transmitted/received first.

#### Cyclic Redundancy Code Field (CRC)

The CRC Field consists of the CRC Sequence (15 bits) and the CRC Delimiter (1 recessive bit). The Cyclic Redundancy Code (CRC) encloses the destuffed bit stream of the Start-of-Frame, Arbitration Field, Control Field, Data Field and CRC Sequence. The most significant bit of the CRC Sequence is transmitted/received first. This frame check sequence, implemented in the CAN Controller is derived from a cyclic redundancy code best suited for frames with a total bit count of less than 127 bytes, CRC Error detection. With Start-of-Frame (dominant bit) included in the code word, any rotation of the code word can be detected by the absence of the CRC Delimiter (recessive bit).

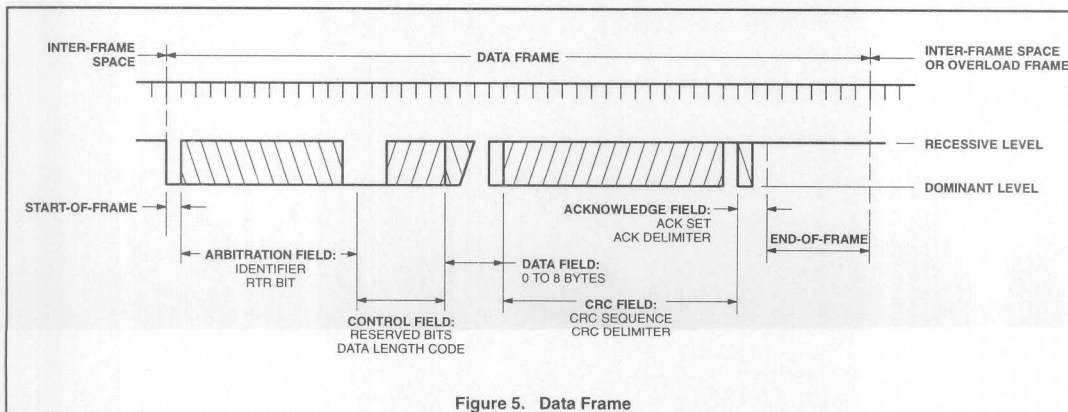


Figure 5. Data Frame

## Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

### Acknowledge Field (ACK)

The Acknowledge Field consists of two bits, the Acknowledge Slot and the Acknowledge Delimiter, which are transmitted with a recessive level by the transmitter of the Data Frame. All CAN Controllers having received the matching CRC Sequence, report this by overwriting the transmitter's recessive bit in the Acknowledge Slot with a dominant bit. Thereby a transmitter, still monitoring the bus level recognizes that at least one receiver within the network has received a complete and correct message (i.e., no error was found). The Acknowledge Delimiter (recessive bit) is the second bit of the Acknowledge Field. As a result, the Acknowledge Slot is surrounded by two recessive bits: the CRC Delimiter and the Acknowledge Delimiter.

All nodes within a CAN network may use all the information coming to the network by all CAN Controllers (shared memory concept). Therefore, acknowledgement and error handling are defined to provide all information in a consistent way throughout this shared memory. Hence, there is no reason to discriminate different receivers of a message in the acknowledge field. If a node is disconnected from the network due to bus failure, this particular node is no longer part of the shared memory. To identify a "lost node" additional and application specific precautions are required.

### End-Of-Frame

Each Data Frame or Remote Frame is delimited by the End-of-Frame bit sequence which consists of seven recessive bits (exceeds the bit stuff width by two bits). Using this method a receiver detects the end of a frame independent of a previous transmission error because the receiver expects all bits up to the end of the CRC Sequence to be coded by the method of bit-stuffing. The bit-stuffing logic is deactivated during the End-of-Frame sequence.

### Remote Frame

A CAN Controller acting as a receiver for certain information may initiate the transmission of the respective data by transmitting a Remote Frame to the network, addressing the data source via the Identifier and setting the RTR bit HIGH (remote; recessive bus level). The Remote Frame is similar to the Data Frame with the following exceptions:

- RTR bit is set HIGH
- Data Length Code is ignored
- No Data Field contained

Note that the value of the Data Length Code should be the one of the corresponding Data Frame, although it is ignored for a Remote Frame.

A Remote Frame is composed of six different bit fields:

- Start-of-Frame
- Arbitration Field
- Control Field
- CRC Field
- Acknowledge Field
- End-of-Frame

### Error Frame

The Error Frame consists of two different fields. The first field is accomplished by the superimposing of Error Flags contributed from different CAN Controllers.

The second field is the Error Delimiter.

#### Error Flag

There are two forms of an Error Flag:

- Active Error Flag, consists of six consecutive dominant bits
- Passive Error Flag, consists of six consecutive recessive bits unless it is overwritten by dominant bits from other CAN Controllers.

An error-active CAN Controller detecting an error condition signals this by transmission of

an Active Error Flag. This Error Flag's form violates the bit-stuffing rule applied to all fields, from Start-of-Frame to CRC Delimiter, or destroys the fixed form of the fields Acknowledge Field or End-of-Frame. Consequently, all other CAN Controllers detect an error condition and start transmission of an Error Flag. Therefore the sequence of dominant bits, which can be monitored on the bus, results from a superposition of different Error Flags transmitted by individual CAN Controllers. The total length of this sequence varies between six (minimum) and twelve (maximum) bits.

An error-passive CAN Controller detecting an error condition tries to signal this by transmission of a Passive Error Flag. The error-passive CAN Controller waits for six consecutive bits with identical polarity, beginning at the start of the Passive Error Flag. The Passive Error Flag is complete when these six identical bits have been detected.

#### Error Delimiter

The Error Delimiter consists of eight recessive bits and has the same format as the Overload Delimiter. After transmission of an Error Flag, each CAN Controller monitors the bus-line until it detects a transition from a dominant-to-recessive bit level. At this point in time, every CAN Controller has finished sending its Error Flag and has additionally sent the first out of the 8 recessive bits of the Error Delimiter. Afterwards all CAN Controllers transmit the remaining recessive bits. After this event and an Intermission Field all error-active CAN Controllers within the network can start a transmission simultaneously.

If a detected error is signaled during transmission of a Data Frame or Remote Frame, the current message is spoiled and a retransmission of the message is initiated.

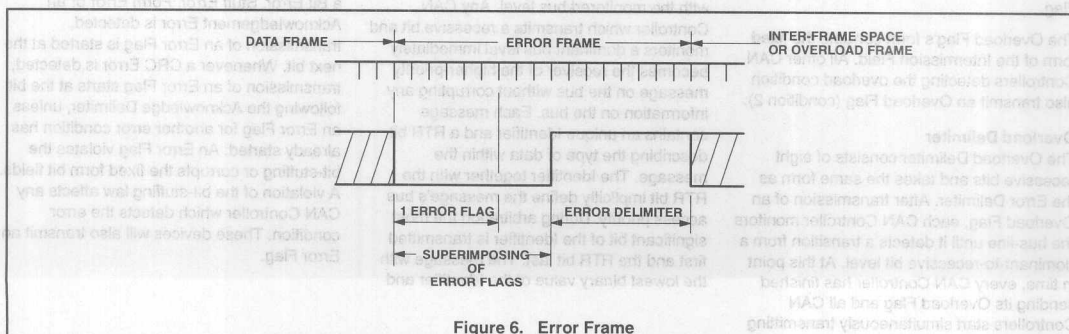


Figure 6. Error Frame

## Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

If a CAN Controller monitors any deviation of the Error Frame, a new Error Frame will be transmitted. Several consecutive Error Frames may result in the CAN Controller becoming error-passive and leaving the network unblocked.

In order to terminate an Error Flag correctly, an error-passive CAN Controller requires the bus to be Bus-Idle for at least three bit periods (if there is a local error at an error-passive-receiver). Therefore a CAN bus should not be permanently loaded.

### Overload Frame

The Overload Frame consists of two fields, the Overload Flag and the Overload Delimiter. There are two conditions in the CAN protocol which lead to the transmission of an Overload Flag:

- Condition 1; receiver circuitry requires more time to process the current data before receiving the next frame (receiver not ready)
- Condition 2; detection of a dominant bit during Intermission Field.

The transmission of an Overload Frame may only start:

- Condition 1; during the first bit period of an expected Intermission Field
- Condition 2; one bit period after detecting the dominant bit during Intermission Field

The 8XC592's on chip CAN Controller will never initiate transmission of a condition 1-Overload Frame and will only react on a transmitted condition 2 Overload Frame, according to the CAN protocol. No more than two Overload Frames are generated to delay a Data Frame or a Remote Frame. Although the overall form of the Overload Frame corresponds to that of the Error Frame, an Overload Frame does not initiate or require the retransmission of the preceding frame.

### Overload Flag

The Overload Flag consists of six dominant bits and has a similar format to the Error Flag.

The Overload Flag's form corrupts the fixed form of the Intermission Field. All other CAN Controllers detecting the overload condition also transmit an Overload Flag (condition 2).

### Overload Delimiter

The Overload Delimiter consists of eight recessive bits and takes the same form as the Error Delimiter. After transmission of an Overload Flag, each CAN Controller monitors the bus-line until it detects a transition from a dominant-to-recessive bit level. At this point in time, every CAN Controller has finished sending its Overload Flag and all CAN Controllers start simultaneously transmitting seven more recessive bits.

### Inter-Frame Space

Data Frames and Remote Frames are separated from preceding frames (all types) by an Inter-Frame Space, consisting of an Intermission Field and a Bus-Idle.

Error-passive CAN Controllers also send a Suspend Transmission after transmission of a message. Overload Frames and Error Frames are not preceded by an Inter-Frame Space.

### Intermission Field

The Intermission Field consists of three recessive bits. During an Intermission period, no frame transmissions will be started by the 8XC592's on chip CAN Controller. An Intermission is required to have a fixed time period to allow a CAN Controller to execute internal processes prior to the next receive or transmit task.

### Bus-Idle

The Bus-Idle time may be of arbitrary length (minimum 0 bit). The bus is recognized to be free and a CAN Controller having information to transmit may access the bus. The detection of a dominant bit level during Bus-Idle on the bus is interpreted as the Start-of-Frame.

### Bus Organization

Bus organization is based on five basic rules described in the following paragraphs.

### Bus Access

CAN Controllers only start transmission during the Bus-Idle state. All CAN Controllers synchronize on the leading edge of the Start-of-Frame (hard synchronization).

### Arbitration

If two or more CAN Controllers simultaneously start transmitting, the bus access conflict is solved by a bit-wise arbitration process during transmission of the Arbitration Field.

During arbitration every transmitting CAN Controller compares its transmitted bit level with the monitored bus level. Any CAN Controller which transmits a recessive bit and monitors a dominant bus level immediately becomes the receiver of the higher-priority message on the bus without corrupting any information on the bus. Each message contains a unique Identifier and a RTR bit describing the type of data within the message. The Identifier together with the RTR bit implicitly define the message's bus access priority. During arbitration the most significant bit of the Identifier is transmitted first and the RTR bit last. The message with the lowest binary value of the Identifier and

RTR bit has the highest priority. A Data Frame has higher priority than a Remote Frame due to its RTR bit having a dominant level.

For every Data Frame there is a unique transmitter. For reasons of compatibility with other CAN-bus controllers, use of the Identifier bit pattern ID = 111111XXXX<sub>b</sub> (X being bits of arbitrary level) is forbidden.

The number of available different Identifiers is 2032 ( $2^{11} - 2^4$ ).

### Coding / Decoding

The following bit fields are coded using the bit-stuffing technique:

- Start-of-Frame
- Arbitration Field
- Control Field
- Data Field
- CRC Sequence

When a transmitting CAN Controller detects five consecutive bits of identical polarity to be transmitted, a complementary (stuff) bit is inserted into the transmitted bit-stream.

When a receiving CAN Controller has monitored five consecutive bits with identical polarity in the received bit streams of the above described bit fields, it automatically deletes the next received (stuff) bit. The level of the deleted stuff bit has to be the complement of the previous bits; otherwise a Stuff Error will be detected and signaled.

The remaining bit fields or frames are of fixed form and are not coded or decoded by the method of bit-stuffing.

The bit-stream in a message is coded according to the Non-Return-to-Zero (NRZ) method, i.e., during a bit period, the bit level is held constant, either recessive or dominant.

### Error Signaling

A CAN Controller which detects an error condition, transmits an Error Flag. Whenever a Bit Error, Stuff Error, Form Error or an Acknowledgement Error is detected, transmission of an Error Flag is started at the next bit. Whenever a CRC Error is detected, transmission of an Error Flag starts at the bit following the Acknowledge Delimiter, unless an Error Flag for another error condition has already started. An Error Flag violates the bit-stuffing or corrupts the fixed form bit fields. A violation of the bit-stuffing law affects any CAN Controller which detects the error condition. These devices will also transmit an Error Flag.

## Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

An error-passive CAN Controller which detects an error condition, transmits a Passive Error Flag. A Passive Error Flag is not able to interrupt a current message at different CAN Controllers but this type of Error Flag may be ignored (overwritten) by other CAN Controllers. After having detected an error condition, an error-passive CAN Controller will wait for six consecutive bits with identical polarity and when monitoring them, interpret them as an Error Flag.

After transmission of an Error Flag, each CAN Controller monitors the bus-line until it detects a transition from a dominant-to-recessive bit level. At this point in time, every CAN Controller has finished transmitting its Error Flag and all CAN Controllers start transmitting seven additional recessive bits.

The message format of a Data Frame or Remote Frame is defined in such a way that all detectable errors can be signaled within the message transmission time and therefore it is very simple for the CAN Controllers to associate an Error Frame to the corresponding message and to initiate retransmission of the corrupted message.

If a CAN Controller monitors any deviation of the fixed form of an Error Frame, it transmits a new Error Frame.

### Overload Signaling

Some CAN Controllers (but not the one on-chip of the 8XC592) require to delay the transmission of the next Data Frame or Remote Frame by transmitting one or more Overload Frames. The transmission of an Overload Frame must start during the first bit of an expected Intermission Field. Transmission of Overload Frames which are reactions on a dominant bit during an expected Intermission Field, start one bit after this event.

Though the format of Overload Frame and Error Frame are identical, they are treated differently. Transmission of an Overload Frame during Intermission Field does not initiate the retransmission of any previous Data Frame or Remote Frame.

If a CAN Controller which transmitted an Overload Frame monitors any deviation of its fixed form, it transmits an Error Frame.

### Error Detection

The processes described in the following paragraphs are implemented in the 8XC592's on-chip CAN Controller for error detection.

#### Bit Error

A transmitting CAN Controller monitors the bus on a bit-by-bit basis. If the bit level

monitored is different from the transmitted one, a Bit Error is signaled. The exceptions being:

- During the Arbitration Field, a recessive bit can be overwritten by a dominant bit. In this case, the CAN Controller interprets this as a loss of arbitration.
- During the Acknowledge Slot, only the receiving CAN Controllers are able to recognize a Bit Error.

#### Stuff Error

The following bit fields are coded using the bit-stuffing technique:

- Start-of-Frame
- Arbitration Field
- Control Field
- Data Field
- CRC Sequence

There are two possible ways of generating a Stuff Error:

- A disturbance generates more than the allowed five consecutive bits with identical polarity. These errors are detected by all CAN Controllers.
- A disturbance falsifies one or more of the five bits preceding the stuff bit. This error situation is not recognized as a Stuff Error by the receivers. Therefore, other error detection processes may detect this error condition such as: CRC check, format violation at the receiving CAN Controllers or Bit Error detection by the transmitting CAN Controller.

#### CRC Error

To ensure the validity of a transmitted message all receivers perform a CRC check. Therefore, in addition to the (destuffed) information digits (Start-of-Frame up to Data Field), every message includes some control digits (CRC Sequence; generated by the transmitting CAN Controller of the respective message) used for error detection.

The code used by all CAN Controllers is a (shortened) BCH code, extended by a parity check and has the following attributes:

- 127 bits as maximum length of the code
- 112 bits as maximum number of information digits (maximum 83 bits are used by the CAN Controller)
- Length of the CRC Sequence amounts to 15 bits
- Hamming distance  $d=6$ .

As a result,  $(d-1)$  random errors are detectable (some exceptions exist).

The CRC Sequence is determined (calculated) by the following procedure.

1. The destuffed bit stream consisting of Start-of-Frame up to the Data Field (if present) is interpreted as polynomial with coefficients 0 or 1.
2. This polynomial is divided (modulo-2) by the following generator polynomial, which includes a parity check:  

$$f(X) = (X^{14} + X^9 + X^8 + X^6 + X^5 + X^4 + X^2 + X + 1) (X + 1) = 1100010110011001_b$$
3. The remainder of this polynomial division is the CRC sequence.

Burst errors are detected up to a length of 15 [degree of  $f(X)$ ]. Multiple errors (number of disturbed bits at least  $d=6$ ) are not detected with a residual error probability of  $2^{-15} (\approx 3 \cdot 10^{-5})$  by CRC check only.

#### Form Error

Form Errors result from violations of the fixed form of the following bit fields:

- CRC Delimiter
- Acknowledge Delimiter
- End-of-Frame
- Error Delimiter
- Overload Delimiter

During the transmission of these bit fields an error condition is recognized if a dominant bit level instead of a recessive one is detected.

#### Acknowledgement Error

This is detected by a transmitter whenever it does not monitor a dominant bit during the Acknowledge Slot.

#### Error Detection by an Error Flag of another CAN Controller

The detection of an error is signaled by transmitting an Error Flag. An Active Error Flag causes a Stuff Error, a Bit Error or a Form Error at all other CAN Controllers.

#### Error Detection Capabilities

Errors which occur at all CAN Controllers (global errors) are 100% detected. For local errors, i.e., for errors occurring at some CAN Controllers only, the shortened BCH code, extended by a parity check, has the following error detection capabilities:

- Up to five single Bit Errors are 100% detected, even if they are distributed randomly within the code
- All single Bit Errors are detected if their total number (within the code) is odd
- The residual error probability of the CRC check amounts to  $3 \cdot 10^{-5}$ .



## Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

### Error Confinement (definitions)

#### Bus-Off

A CAN Controller which has too many unsuccessful transmissions, relative to the number of successful transmissions, will enter the Bus-Off state. It remains in this state, neither receiving nor transmitting messages until the Reset Request bit is set LOW (absent) and both Error Counters set to 0.

#### Acknowledge (ACK)

A CAN Controller which has received a valid message correctly, indicates this to the transmitter by transmitting a dominant bit level on the bus during the Acknowledge Slot, independent of accepting or rejecting the message.

#### Error-Active

An error-active CAN Controller in its normal operating state is able to receive and to transmit normally and also to transmit an Active Error Flag.

#### Error-Passive

An error-passive CAN Controller may transmit or receive messages normally. In the case of a detected error condition it transmits a Passive Error Flag instead of an Active Error Flag.

Hence the influence on bus activities by an error-active CAN Controller (e.g., due to a malfunction) is reduced.

#### Suspend Transmission

After an error-passive CAN Controller has transmitted a message, it sends eight recessive bits after the Intermission Field and then checks for Bus-Idle. If during Suspend Transmission another CAN Controller starts transmitting a message the suspended CAN

Controller will become the receiver of this message; otherwise being in Bus-Idle it may start to transmit a further message.

#### Start-Up

A CAN Controller which either was switched off or is in the Bus-Off state, must run a Start-Up routine in order to:

- Synchronize with other available CAN Controllers before starting to transmit. Synchronization is achieved, when 11 recessive bits, equivalent to Acknowledge Delimiter, End-of-Frame and Intermission Field, have been detected (Bus-Free).
- Wait for other CAN Controllers without passing into the Bus-Off state (due to a missing acknowledge), if there is no other CAN Controller currently available.

### Aims of Error Confinement

#### Distinction of Short and Long Lasting Disturbances

The CPU must be informed when there are long-lasting disturbances and when bus activities have returned to normal operation.

During long-lasting disturbances, a CAN Controller enters the Bus-Off state and the CPU may use default values.

Minor disturbances of bus activities will not affect a CAN Controller. In particular, a CAN Controller does not enter the Bus-Off state or inform the CPU of a short-lasting bus disturbance.

#### Detection and Localization of Hardware Disturbance and Defects

The rules for error confinement are defined by the CAN protocol specification (and implemented in the 8XC592's on-chip CAN Controller), in such a way that the CAN Controller, being nearest to the error-locus,

reacts with a high probability the quickest (i.e., becomes error-passive or Bus-Off). Hence errors can be localized and their influence on normal bus activities is minimized.

#### Error Confinement

All CAN Controllers contain a Transmit Error Counter and a Receive Error Counter, which registers errors during the transmission and the reception of messages, respectively.

If a message is transmitted or received correctly, the count is decreased. In the event of an error, the count is increased. The Error Counters have a non-proportional method of counting: an error causes a larger counter increase than a correctly transmitted/received message causes the count to decrease. Over a period of time this may result in an increase in error counts, even if there are fewer corrupted messages than uncorrupted ones. The level of the Error Counters reflect the relative frequency of disturbances. The ratio of increase/decrease depends on the acceptable ratio of invalid/valid messages on the bus and is hardware implemented to eight.

If one of the Error Counters exceeds the Warning Limit of 96 error points, indicating a significant accumulation of error conditions, this is signaled by the CAN Controller (Error-Status, Error Interrupt).

A CAN Controller operates in the error-active mode until it exceeds 127 error points on one of its Error Counters. At this point it will enter the error-passive state.

A transmit error which exceeds 255 error points results in the CAN Controller entering the Bus-Off state.

Form Error at all other CAN Controllers

Error Detection Capabilities

Errors which occur at all CAN Controllers (global errors) are 100% detected. For local errors, i.e., for errors occurring at some CAN Controller only, the enhanced BCH code, extended by a parity check, has the following error detection capabilities:

- Up to five single bit errors are 100% detected, even if they are distributed randomly within the code
- All single bit errors are detected if their total number (within the code) is odd
- The random error probability of the CRC check amounts to  $2^{-16}$

messages used for error detection.

The codes used by all CAN Controllers are a (shortened) BCH code, extended by a parity check and has the following attributes:

- 127 bits as maximum length of the code
- 112 bits as maximum number of information bits (maximum 83 bits are used by the CAN Controller)
- Length of the CRC sequence amounts to 15 bits
- Hamming distance  $d=8$

As a result, (4-1) random errors are detectable (some exceptions exist).

The CRC sequence is determined (estimated) by the following procedure:

Through the format of Overlaid Frames and Error Frames are indicated, they are treated differently. Transmission of an Overlaid Frame during Intermission Field does not initiate the termination of any previous Data Frame or Remote Frame.

If a CAN Controller which transmitted an Overlaid Frame monitors any deviation of its Data Frame or Remote Frame, it transmits an Error Frame.

#### Error Detection

The procedure described in the following paragraphs was implemented in the 8XC592's on-chip CAN Controller for error detection.

Bit Error

A transmitting CAN Controller monitors the bus on a bit-by-bit basis. If the bit level



## Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

### INTERRUPT SYSTEM

External events and the real-time-driven on-chip peripherals require service by the CPU asynchronously to the execution of any particular section of code. To tie the asynchronous activities of these functions to normal program execution a multiple-source, two-priority-level, nested interrupt system is provided. Interrupt response latency is from 2.25µs to 7.5µs when using a 16MHz crystal. The latency time depends on the sequence of instructions executed directly after an interrupt request. During a CAN-DMA transfer the interrupt system is disabled. The 8XC592 acknowledges interrupt requests from fifteen sources as follows:

- INT0 and INT1: externally via pins 27 and 28 respectively
- Timer 0 and Timer 1: from the two internal counters
- Timer T2 (8 separate interrupts): 4 capture interrupts, 3 compare interrupts and an overflow interrupt
- ADC end-of-conversion interrupt
- CAN Controller interrupt
- UART serial I/O port interrupt.

#### Interrupt Enable Registers

IEN0 (A8H)							
7	6	5	4	3	2	1	0
EA	EAD	ES1	ES0	ET1	EX1	ET0	EX0

Bit	Symbol	Function
IEN0.7	EA	General enable/disable control  0 = No interrupt is enabled 1 = Any individually enabled interrupt will be accepted
IEN0.6	EAD	Enable ADC interrupt
IEN0.5	ES1	Enable SIO1 (CAN) interrupt
IEN0.4	ES0	Enable SIO0 (UART) interrupt
IEN0.3	ET1	Enable Timer 1 interrupt
IEN0.2	EX1	Enable External 1 interrupt
IEN0.1	ET0	Enable Timer 0 interrupt
IEN0.0	EX0	Enable External 0 interrupt

IEN1 (E8H)							
7	6	5	4	3	2	1	0
ET2	ECM2	ECM1	ECM0	ECT3	ECT2	ECT1	ECT0

Bit	Symbol	Function
IEN1.7	ET2	Enable T2 overflow interrupt(s)
IEN1.6	ECM2	Enable T2 comparator 2 interrupt
IEN1.5	ECM1	Enable T2 comparator 1 interrupt
IEN1.4	ECM0	Enable T2 comparator 0 interrupt
IEN1.3	ECT3	Enable T2 capture register 3 interrupt
IEN1.2	ECT2	Enable T2 capture register 2 interrupt
IEN1.1	ECT1	Enable T2 capture register 1 interrupt
IEN1.0	ECT0	Enable T2 capture register 0 interrupt

Where: 0 = interrupt disabled.  
1 = interrupt enabled.

IP0 (B8H)							
7	6	5	4	3	2	1	0
—	PAD	PS1	PS0	PT1	PX1	PT0	PX0

Bit	Symbol	Function
IP0.7	—	Unused
IP0.6	PAD	ADC interrupt priority level
IP0.5	PS1	SIO1 (CAN) interrupt priority level
IP0.4	PS0	SIO0 (UART) interrupt priority level
IP0.3	PT1	Timer 1 interrupt priority level
IP0.2	PX1	External interrupt 1 priority level
IP0.1	PT0	Timer 0 interrupt priority level
IP0.0	PX0	External interrupt 0 priority level

IP1 (F8H)							
7	6	5	4	3	2	1	0
PT2	PCM2	PCM1	PCM0	PCT3	PCT2	PCT1	PCT0

Bit	Symbol	Function
IP1.7	PT2	T2 overflow interrupt(s) priority level
IP1.6	PCM2	T2 comparator 2 priority interrupt level
IP1.5	PCM1	T2 comparator 1 priority interrupt level
IP1.4	PCM0	T2 comparator 0 priority interrupt level
IP1.3	PCT3	T2 capture register 3 priority interrupt level
IP1.2	PCT2	T2 capture register 2 priority interrupt level
IP1.1	PCT1	T2 capture register 1 priority interrupt level
IP1.0	PCT0	T2 capture register 0 priority interrupt level

Where: 0 = interrupt disabled.  
1 = interrupt enabled.

Table 10 shows the interrupt vectors. The vector indicates the Program memory location where the appropriate interrupt service routine starts.

**Table 10. Interrupt Vectors**

SOURCE		VECTOR
External 0	X0	0003H
Timer 0 overflow	T0	000BH
External 1	X1	0013H
Timer 1 overflow	T1	001BH
Serial I/O 0 (UART)	S0	0023H
Serial I/O 1 (CAN)	S1	002BH
T2 capture 0	CT0	0033H
T2 capture 1	CT1	003BH
T2 capture 2	CT2	0043H
T2 capture 3	CT3	004BH
ADC completion	ADC	0053H
T2 compare 0	CM0	005BH
T2 compare 1	CM1	0063H
T2 compare 2	CM2	006BH
T2 overflow	T2	0073H

#### Interrupt Priority

Each interrupt source can be either high priority or low priority. If both priorities are requested simultaneously, the processor will branch to the high priority vector. If there are simultaneous requests from sources of the same priority, then interrupts will be serviced in the following order: X0, S1, ADC, T0, CT0, CM0, X1, CT1, CM1, T1, CT2, CM2, S0, CT3, T2.

A low priority interrupt routine can only be interrupted by a high priority interrupt. A high priority interrupt routine cannot be interrupted.

# Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

Table 11. Status of External Pins During Sleep, Idle and Power-Down Modes

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3	PORT 4	PWM0/PWM1
Idle	Internal	1	1	Port Data	Port Data	Port Data	Port Data	Port Data	High
Idle	External	1	1	Float	Port Data	Address	Port Data	Port Data	High
Power-down	Internal	0	0	Port Data	Port Data	Port Data	Port Data	Port Data	High
Power-down	External	0	0	Float	Port Data	Port Data	Port Data	Port Data	High

**NOTE:**

If the port pins P1.6 and P1.7 are used as the CAN transmitter outputs (CTX0 and CTX1), then during Sleep and Power-down mode these pins output a "recessive" level.

**POWER REDUCTION MODES**

The 8XC592 has three software-selectable modes to reduce power consumption. These are:

- Sleep mode, affecting the CAN Controller only
- Idle mode, affecting the
  - CPU (halted)
  - Timer 2 (stopped and reset)
  - PWM0, PWM1 (reset, output = HIGH)
  - ADC (aborted if in progress)
- Power-down mode, affecting the whole 8XC592 device.

**CAN Sleep Mode**

In order to reduce power consumption of the P8XC592 the CAN Controller may be switched off (disconnecting the internal clock) by setting the CAN Command Register bit 4 (Sleep) HIGH. The CAN Controller leaves this Sleep mode by detecting either activity on the CAN-bus (dominant bit-level on CRX0/CXR1) or by setting the Sleep bit to LOW.

As the CPU cannot only write to the Sleep bit, but can also read it, the CAN Controller status can be determined directly.

**Idle Mode**

The instruction that sets PCON.0 is the last one executed in the normal operating mode before Idle mode is activated. Once in the Idle mode, the CPU status is preserved in its entirety: the Stack Pointer, Program Counter, Program Status Word, Accumulator, RAM and all other registers maintain their data during Idle mode. The status of the external pins during Idle mode is shown in Table 12. There are three ways to terminate the Idle mode:

Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, provided that the interrupt source is active during Idle

mode. After the interrupt is serviced, the program continues with the instruction immediately after the one, at which the interrupt request was detected.

The flag bits GF0 and GF1 may be used to determine whether the interrupt was received during normal execution or during the Idle mode. For example, the instruction that writes to PCON.0 can also set or clear one or both flag bits. When Idle mode is terminated by an interrupt, the service routine can examine the status of the flag bits.

Another way of terminating the Idle mode is an external hardware reset. Since the oscillator is still running, the reset signal is required to be active only for 2 machine cycles (24 oscillator periods) to complete the reset operation.

The third way is the internally generated watchdog reset after an overflow of Timer 3.

**Power-down Mode**

The instruction that sets PCON.1 to HIGH, is the last one executed before entering the Power-down mode. In Power-down mode the oscillator of the P8XC592 is stopped. If the CAN Controller is in use, it is recommended to set it into Sleep mode before entering Power-down mode. However, setting PCON.1 to HIGH also sets the Sleep bit (CAN Controller Command Register bit 4) to HIGH.

The P8XC592 leaves Power-down mode either by a hardware reset or by a CAN Wake-up interrupt (due to activity on the CAN bus), if the SI01 (CAN) interrupt source is enabled (contents of register IEN0 = 1x1xxxxb). A hardware reset affects the whole P8XC592, but leaves the contents of the on-chip RAM unchanged (CAN Controller and CPU's Special Function Registers are reset). A CAN Wake-up interrupt during Power-down mode causes a reset output

pulse with a width of 6144 machine cycles (4.6ms with  $f_{CLK}=16\text{MHz}$ ). All hardware except from the CAN Controller of the P8XC592 is reset (the contents of all CAN Controller registers are preserved).

Note that a capacitance connected to the RST pin could lengthen the internally generated reset pulse. If the pulse exceeds 8192 machine cycles, the CAN Controller part is reset too.

**RESET**

A reset is accomplished by holding the RST pin HIGH for at least two machine cycles (24 oscillator periods). The CPU responds by executing an internal reset. During reset ALE and PSEN output at a HIGH level. In order to perform a correct reset, this level must not be affected by external elements.

Also with the P8XC592, the RST line can be pulled HIGH internally by a pull-up transistor activated by the watchdog timer T3. The length of the output pulse from T3 is 3 machine cycles. A pulse of such short duration is necessary in order to recover from a processor or system fault as fast as possible.

During Power-down a reset could be generated internally via the CAN Wake-up interrupt. Then the RST pin is pulled HIGH for 6144 machine cycles. In this case the CAN Controller is not reset.

If the watchdog timer or the CAN Wake-up interrupt is used to reset external devices, the usual capacitor arrangement for power-on reset should not be used. However, the internal reset is forced, independent of the external level on the RST pin.

The Main RAM and AuxRAM are not affected. When VDD is turned on, the RAM content is indeterminate.

# Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

**Table 12.**

After a reset the internal registers have the following contents:

(CPU PART)	REGISTER	7	6	5	4	3	2	1	0
	ACC	0	0	0	0	0	0	0	0
	ADCO	X	X	0	0	0	0	0	0
	ADCH	X	X	X	X	X	X	X	X
	B	0	0	0	0	0	0	0	0
	CML0 – CML2	0	0	0	0	0	0	0	0
	CMH0 – CMH2	0	0	0	0	0	0	0	0
	CTCON	0	0	0	0	0	0	0	0
	CTL0 – CTL3	X	X	X	X	X	X	X	X
	CTH0 – CTH3	X	X	X	X	X	X	X	X
	DPL	0	0	0	0	0	0	0	0
	DPH	0	0	0	0	0	0	0	0
	IEN0	0	0	0	0	0	0	0	0
	IEN1	0	0	0	0	0	0	0	0
	IP0	X	0	0	0	0	0	0	0
	IP1	0	0	0	0	0	0	0	0
	PCH	0	0	0	0	0	0	0	0
	PCL	0	0	0	0	0	0	0	0
	PCON	0	X	X	0	0	0	0	0
	PSW	0	0	0	0	0	0	0	0
	PWM0	0	0	0	0	0	0	0	0
	PWM1	0	0	0	0	0	0	0	0
	PWMP	0	0	0	0	0	0	0	0
	P0 – P4	1	1	1	1	1	1	1	1
	P5	X	X	X	X	X	X	X	X
	RTE	0	0	0	0	0	0	0	0
	SOBUF	X	X	X	X	X	X	X	X
	SOCON	0	0	0	0	0	0	0	0
	CANSTA	0	0	0	0	1	1	0	0
	CANCON	X	X	X	0	0	0	0	0
	CANDAT	X	X	X	X	X	X	X	X
	CANADR	0	X	1	0	0	1	0	0
	SP	0	0	0	0	0	1	1	1
	STE	1	1	0	0	0	0	0	0
	TCON	0	0	0	0	0	0	0	0
	TH0, TH1	0	0	0	0	0	0	0	0
	TMH2	0	0	0	0	0	0	0	0
	TL0, TL1	0	0	0	0	0	0	0	0
	TML2	0	0	0	0	0	0	0	0
	TMOD	0	0	0	0	0	0	0	0
	TM2CON	0	0	0	0	0	0	0	0
	TM2IR	0	0	0	0	0	0	0	0
	T3	0	0	0	0	0	0	0	0
(CAN PART)	REGISTER	7	6	5	4	3	2	1	0
	CR	0	X	1	X	X	X	X	1
	CMR	1	1	X	0	X	X	X	X
	SR	0	0	0	0	1	1	0	0
	IR	X	X	X	0	0	0	0	0
	ACR	X	X	X	X	X	X	X	X
	AMR	X	X	X	X	X	X	X	X
	BTR 0	X	X	X	X	X	X	X	X
	BTR 1	X	X	X	X	X	X	X	X
	OCR	X	X	X	X	X	X	X	X
	TR	X	X	X	X	X	X	X	X
	TXB 10 – 19	X	X	X	X	X	X	X	X
	RXB 20 – 29	X	X	X	X	X	X	X	X

**NOTE:**

X = Undefined State

Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

**ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>**

PARAMETER	RATING		UNIT
	MIN	MAX	
Storage temperature range	-65	+150	°C
Voltage on EA/V <sub>PP</sub> to V <sub>SS</sub>	-0.5	+13	V
Power dissipation (based on package heat transfer limitations, not device power consumption)	1.0		W
Voltage on V <sub>DD</sub> pin	-0.5	+6.5	V
Input voltage on any pin except from CTX0, CTX1, CRX0 and CRX1	-0.5	V <sub>DD</sub> +0.5	V
Input output current on any I/O pin except fro CTX0 and CTX1	—	10	mA
Sink current of CTX0, CTX1 together	—	30	mA
Source current of CTX0, CTX1 together	—	20	mA

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
2. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
3. Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to  $V_{SS}$  unless otherwise noted.

# Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

## DC ELECTRICAL CHARACTERISTICS

$T_{amb} = -40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$  (83C592FFA, 87C592EFA),  $T_{amb} = -40^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$  (83C592FHA);  $V_{DD}, AV_{DD} = 5\text{V} \pm 10\%$ ;  $V_{SS}, AV_{SS} = 0\text{V}$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYPICAL	MAX	
$V_{DD}$	Supply voltage		4.5		5.5	V
$I_{DD}$	Supply current:					
$I_{ID}$	operating	$f_{CLK} = 16\text{MHz}$	—		50	mA
$I_{IS}$	Idle mode	$f_{CLK} = 16\text{MHz}$	—		15	mA
$I_{PD}$	Idle and Sleep mode	$f_{CLK} = 16\text{MHz}$	—		10	mA
$I_{PD}$	Power-down mode (83C592 FFA)	Note 4			150	$\mu\text{A}$
$I_{PD}$	Power-down mode (8XC592 xFx)	Note 4			100	$\mu\text{A}$
<b>Inputs</b>						
$V_{IL}$	Input low voltage, except $\overline{\text{EA}}$ , CRX0, CRX1		-0.5		$0.2V_{DD}-0.1$	V
$V_{IL1}$	Input low voltage to $\overline{\text{EA}}$		-0.5		$0.2V_{DD}-0.3$	V
$V_{IH}$	Input high voltage, except XTAL1, RST, CRX0, CRX1		$0.2V_{DD}+0.9$		$V_{DD}+0.5$	V
$V_{IH1}$	Input high voltage, XTAL1, RST		$0.7V_{DD}$		$V_{DD}+0.5$	V
$-I_{IL}$	Input current Low, ports 1, 2, 3, 4	$V_{IN} = 0.45\text{V}$			-50	$\mu\text{A}$
$-I_{TL}$	High-to-Low transition current, ports 1, 2, 3, 4	$V_I = 2.0\text{V}/0.45\text{V}$			-650	$\mu\text{A}$
$\pm I_{IL1}$	Input leakage current, port 0, $\overline{\text{EA}}$ , STADC, EW	$0.45\text{V} < V_I < V_{DD}$			10	$\mu\text{A}$
$\pm I_{IL2}$	Input leakage current, port 5	$0.45\text{V} < V_I < V_{DD}$			1	$\mu\text{A}$
<b>Outputs</b>						
$V_{OL}$	Output low voltage, ports 1, 2, 3, 4, except P1.6, P1.7 <sup>5,6</sup>	$I_{OL} = 1.6\text{mA}^2$			0.45	V
$V_{OL1}$	Output low voltage, port 0, ALE, PSEN, PWM0, PWM1, P1.6, P1.7 <sup>5,6</sup>	$I_{OL} = 3.2\text{mA}^2$			0.45	V
$V_{OH}$	Output high voltage, ports 1, 2, 3, 4	$-I_{OH} = 60\mu\text{A}$ $-I_{OH} = 25\mu\text{A}$ $-I_{OH} = 10\mu\text{A}$	2.4 $0.75V_{DD}$ $0.9V_{DD}$			V V V
$V_{OH1}$	Output high voltage (port 0 in external bus mode, ALE, PSEN, PWM0, PWM1) <sup>3</sup>	$-I_{OH} = 400\mu\text{A}$ $-I_{OH} = 150\mu\text{A}$ $-I_{OH} = 40\mu\text{A}$	2.4 $0.75V_{DD}$ $0.9V_{DD}$			V V V
$V_{OH2}$	High level output voltage (RST)	$-I_{OH} = 400\mu\text{A}$ $-I_{OH} = 120\mu\text{A}$	2.4 $0.8V_{DD}$			V
$R_{RST}$	Internal reset pull-down resistor		50		150	$\text{k}\Omega$
$C_{IO}$	I/O buffer pin capacitance	Test freq = 1MHz, $T_{amb} = 25^{\circ}\text{C}$			10	pF

$V_{OL2}$	Output low voltage, port 5	$I_{OL} = 1.6\text{mA}^2$			0.45	V
$V_{OL3}$	Output low voltage, port 0, ALE, PSEN, PWM0, PWM1, P1.6, P1.7 <sup>5,6</sup>	$I_{OL} = 3.2\text{mA}^2$			0.45	V
$V_{OH3}$	Output high voltage, port 5	$-I_{OH} = 60\mu\text{A}$ $-I_{OH} = 25\mu\text{A}$ $-I_{OH} = 10\mu\text{A}$	2.4 $0.75V_{DD}$ $0.9V_{DD}$			V V V
$V_{OH4}$	Output high voltage, port 0, ALE, PSEN, PWM0, PWM1	$-I_{OH} = 400\mu\text{A}$ $-I_{OH} = 150\mu\text{A}$ $-I_{OH} = 40\mu\text{A}$	2.4 $0.75V_{DD}$ $0.9V_{DD}$			V V V
$V_{OH5}$	High level output voltage (RST)	$-I_{OH} = 400\mu\text{A}$ $-I_{OH} = 120\mu\text{A}$	2.4 $0.8V_{DD}$			V
$R_{RST}$	Internal reset pull-down resistor		50		150	$\text{k}\Omega$
$C_{IO}$	I/O buffer pin capacitance	Test freq = 1MHz, $T_{amb} = 25^{\circ}\text{C}$			10	pF



# Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

## DC ELECTRICAL CHARACTERISTICS (Continued)

$T_{amb} = -40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$  (83C592FFA, 87C592EFA),  $T_{amb} = -40^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$  (83C592FHA);  $V_{DD}$ ,  $AV_{DD} = 5\text{V} \pm 10\%$ ,  $V_{SS}$ ,  $AV_{SS} = 0\text{V}$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYPICAL	MAX	
Analog Inputs						
AV <sub>DD</sub>	Analog supply voltage <sup>7</sup>	AV <sub>DD</sub> = V <sub>DD</sub> ±0.2V	4.5		5.5	V
AI <sub>DD</sub>	Analog supply current: Operating Idle mode	Port 5 = AV <sub>DD</sub>			2.5 2.5	 mA mA
AI <sub>IS</sub>	Idle and sleep mode	83C592FHA 8XC592xFx			400 350	 μA μA
AI <sub>PD</sub>	Power-down mode	83C592FHA 8XC592xFx			400 350	 μA μA
AV <sub>IN</sub>	Analog input voltage		AV <sub>SS</sub> -0.2		AV <sub>DD</sub> +0.2	V
AV <sub>REF</sub>	Reference voltage: AV <sub>REF-</sub> AV <sub>REF+</sub>		AV <sub>SS</sub> -0.2		AV <sub>DD</sub> +0.2	 V V
R <sub>REF</sub>	Resistance between AV <sub>REF+</sub> and AV <sub>REF-</sub>		10		50	kΩ
C <sub>IA</sub>	Analog input capacitance				15	pF
t <sub>ADS</sub>	Sampling time				8t <sub>CY</sub>	μs
t <sub>ADC</sub>	Conversion time (including sampling time)				50t <sub>CY</sub>	μs
DL <sub>e</sub>	Differential non-linearity <sup>8, 9, 10</sup>				±1	LSB
IL <sub>e</sub>	Integral non-linearity <sup>8, 11</sup>				±2	LSB
OS <sub>e</sub>	Offset error <sup>8, 12</sup>				±2	LSB
G <sub>e</sub>	Gain error <sup>8, 13</sup>				±0.4	%
A <sub>e</sub>	Absolute Voltage Error <sup>8, 14</sup>				±3	LSB
M <sub>CTC</sub>	Channel to channel matching				±1	LSB
C <sub>t</sub>	Crosstalk between Port 5 inputs <sup>9</sup>	0–100kHz			–60	dB
CAN						
	CAN input comparator (CRX0, CRX1)	AV <sub>DD</sub> = 5V ± 5% 1.4V < V <sub>I</sub> < AV <sub>DD</sub> –1.4V				
±V <sub>DIF</sub>	Differential input voltage <sup>15</sup>		32		–	mV
V <sub>HYST</sub>	Hysteresis voltage <sup>15</sup>		8		30	mV
±I <sub>I</sub>	Input current		–		400	nA
	CAN output driver	V <sub>DD</sub> = 5V ± 5%				
V <sub>OLT</sub>	CTX0, CTX1, output voltage LOW	I <sub>O</sub> = 1.2mA <sup>15</sup> I <sub>O</sub> = 10mA	– –		0.1 0.6	 V V
V <sub>OHT</sub>	CTX0, CTX1, output voltage HIGH	I <sub>O</sub> = 1.2mA <sup>15</sup> I <sub>O</sub> = –10mA <sup>16</sup>	V <sub>DD</sub> – 0.1 V <sub>DD</sub> – 0.6		– –	 V V
Reference						
V <sub>REFOUT</sub>	Reference REF output voltage <sup>15</sup> (bit Reference Active = HIGH)	AV <sub>DD</sub> = 5V ± 5% –0.1mA < I <sub>L</sub> < 0.1mA; C <sub>L</sub> = 10nF	AV <sub>DD</sub> /2–0.05		AV <sub>DD</sub> /2+0.05	V
±I <sub>REFIN</sub>	REF input current (bit Reference Active = LOW)	1.5V < V <sub>REFIN</sub> < AV <sub>DD</sub> –1.5V	–		10	μA

NOTES ON NEXT PAGE.

### Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

## NOTES TO THE DC ELECTRICAL CHARACTERISTICS:

1. The operating current is measured with all output pins unloaded; XTAL1 is driven with  $t_R = t_F = 10\text{ns}$ ;  $V_{IL} = V_{SS} + 0.5V$ ;  $V_{IH} = V_{DD} - 0.5V$ ;  $\overline{EA} = \overline{RST} = \text{Port } 0 = P1.6 = P1.7 = \overline{EW} = V_{DD}$ ;  $\overline{STADC} = V_{SS}$ ;  $\overline{CRX0} = 2.7V$ ;  $\overline{CRX1} = 2.3V$ .
2. The idle mode supply current is measured with all output pins unloaded; XTAL1 is driven with  $t_R = t_F = 10\text{ns}$ ;  $V_{IL} = V_{SS} + 0.5V$ ;  $V_{IH} = V_{DD} - 0.5V$ ;  $\text{Port } 0 = P1.6 = P1.7 = \overline{EW} = V_{DD}$ ;  $\overline{EA} = \overline{RST} = \overline{STADC} = V_{SS}$ ;  $\overline{CRX0} = 2.7V$ ;  $\overline{CRX1} = 2.3V$ .
3. The idle and sleep mode supply current is measured with all output pins unloaded; XTAL1 is driven with  $t_R = t_F = 10\text{ns}$ ;  $V_{IL} = V_{SS} + 0.5V$ ;  $V_{IH} = V_{DD} - 0.5V$ ;  $\text{Port } 0 = P1.6 = P1.7 = \overline{EW} = \overline{CRX0} = V_{DD}$ ;  $\overline{EA} = \overline{RST} = \overline{STADC} = \overline{CRX1} = V_{SS}$ ;  $\text{CAN: register } 6 = 00H, \text{ register } 7 = 12H$ ;  $\text{register } 8 = 02H, \text{ register } 0 = 20H, \text{ wait } 15t_{CY}, \text{ register } 1 = 10H, \text{ wait for bit Sleep} = 1$ .
4. The power-down current is measured with all output pins unloaded;  $\text{Port } 0 = P1.6 = P1.7 = \overline{EW} = \overline{CRX0} = V_{DD}$ ;  $\overline{XTAL1} = \overline{EA} = \overline{RST} = \overline{STADC} = \overline{CRX1} = V_{SS}$ . Windowed devices must have the window covered during testing.
5. Under steady state (non-transient) conditions,  $I_{OL}$  must be limited externally as follows:
 

Maximum $I_{OL}$ per 8 bit port	
– Port 0:	26mA
– Port 1:	32mA
– Ports 2, 3, and 4:	15mA
Maximum $I_{OL}$ for all output pins:	
71mA	

If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
6. Capacitive loads on Port0 and Port2 may degrade the LOW level output voltage of ALE, Port1 and Port2. During a 1-to-0 transition on the Port0 and Port2 pins and a capacitive load  $> 100\text{pF}$ , the ALE LOW level may exceed 0.8V. In that case, it is necessary to connect ALE to a Schmitt trigger input respectively use an address latch with a Schmitt trigger STROBE input.
7. Capacitive loads on Port0 and Port2 may cause a HIGH level output voltage degradation of ALE and  $\overline{PSEN}$  below 0.9  $V_{DD}$  during the address bits are stabilizing.
8.  $AV_{REF+} = 5.12V$ ;  $AV_{REF-} = 0V$ ;  $AV_{DD} = 5.0V$ .
9. The differential non-linearity ( $DL_n$ ) is the difference between the actual step width and the ideal step width.
10. The ADC is monotonic, there are no missing codes.
11. The integral non-linearity ( $IL_n$ ) is the peak difference between the center of the steps of the actual and the ideal transfer curve after appropriate adjustment of gain and offset error.
12. The offset error ( $OS_n$ ) is the absolute difference between the straight line which fits the actual transfer curve after removing gain error, and a straight line which fits the ideal transfer curve. The offset error is constant at every point of the actual transfer curve.
13. The gain error ( $G_n$ ) is the relative difference in percent between the straight line fitting the actual transfer curve after removing offset error and the straight line which fits the ideal transfer curve. The gain error is constant at every point on the transfer curve.
14. The absolute voltage error ( $A_n$ ) is the maximum difference between the center of the steps of the actual transfer curve of the not calibrated ADC and the ideal transfer curve.
15. Not tested during production.
16. Source current for the CTX0, CTX1 outputs together.

# Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

## AC ELECTRICAL CHARACTERISTICS

 $T_{amb} = -40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $V_{DD}$ ,  $AV_{DD} = 5V \pm 10\%$ ,  $V_{SS}$ ,  $AV_{SS} = 0V$ 

SYMBOL	FIGURE	PARAMETER	VARIABLE CLOCK		UNIT
			MIN	MAX	
$f_{CLK}$	7	Oscillator frequency	1.2	16	MHz
$t_{LHLL}$	7	ALE pulse width	$2t_{CLK}-40$		ns
$t_{AVLL}$	7	Address valid to ALE low	$t_{CLK}-55$		ns
$t_{LLAX}$	7	Address hold after ALE low	$t_{CLK}-35$		ns
$t_{LLIV}$	7	ALE low to valid instruction in		$4t_{CLK}-100$	ns
$t_{LLPL}$	7	ALE low to PSEN low	$t_{CLK}-40$		ns
$t_{PLPH}$	7	PSEN pulse width	$3t_{CLK}-45$		ns
$t_{PLIV}$	7	PSEN low to valid instruction in		$3t_{CLK}-105$	ns
$t_{PXIX}$	7	Input instruction hold after PSEN	0		ns
$t_{PXIZ}$	7	Input instruction float after PSEN		$t_{CLK}-25$	ns
$t_{AVIV}$	7	Address to valid instruction in		$5t_{CLK}-105$	ns
$t_{PLAZ}$	7	PSEN low to address float		10	ns
<b>Data Memory</b>					
$t_{AVLL}$	8, 9	Address valid to ALE low	$t_{CLK}-55$		ns
$t_{RLRH}$	8, 9	RD pulse width	$6t_{CLK}-100$		ns
$t_{WLWH}$	8, 9	WR pulse width	$6t_{CLK}-100$		ns
$t_{RLDV}$	8, 9	RD low to valid data in		$5t_{CLK}-165$	ns
$t_{RHDX}$	8, 9	Data hold after RD	0		ns
$t_{RHDZ}$	8, 9	Data float after RD		$2t_{CLK}-70$	ns
$t_{LLDV}$	8, 9	ALE low to valid data in		$8t_{CLK}-150$	ns
$t_{AVDV}$	8, 9	Address to valid data in		$9t_{CLK}-165$	ns
$t_{LLWL}$	8, 9	ALE low to RD or WR low	$3t_{CLK}-50$		ns
$t_{AVWL}$	8, 9	Address valid to WR low or RD low	$4t_{CLK}-130$		ns
$t_{QVWX}$	8, 9	Data valid to WR transition	$t_{CLK}-60$		ns
$t_{WHQX}$	8, 9	Data hold after WR	$t_{CLK}-50$		ns
$t_{RLAZ}$	8, 9	RD low to address float		0	ns
$t_{WHLH}$	8, 9	RD or WR high to ALE high	$t_{CLK}-40$	$t_{CLK}+40$	ns
<b>External Clock</b>					
$t_{CHCX}$	11	High time <sup>3</sup>	20		ns
$t_{CLCX}$	11	Low time <sup>3</sup>	20		ns
$t_{CLCH}$	11	Rise time <sup>3</sup>		20	ns
$t_{CHCL}$	11	Fall time <sup>3</sup>		20	ns
<b>UART Timing in Shift Register Mode</b>					
$t_{XLXL}$	10	Serial port clock cycle time <sup>3</sup>	$12t_{CLK}$		$\mu\text{s}$
$t_{QVXH}$	10	Output data setup to clock rising edge	$10t_{CLK}-133$		ns
$t_{XHGX}$	10	Output data hold after clock rising edge	$2t_{CLK}-117$		ns
$t_{XHDX}$	10	Input data hold after clock rising edge	0		ns
$t_{XHDV}$	10	Clock rising edge to input data valid		$10t_{CLK}-133$	ns
<b>CAN Input Comparator / Output Driver (<math>AV_{DD} = 5V \pm 5\%</math>)</b>					
$t_{SD}$		Sum of input and output delay ( $V_{DIF} = \pm 32\text{mV}$ ; $1.4V < V_I < AV_{DD} - 1.4V$ ) $AV_{DD} = 5V \pm 5\%$	—	60	ns

### NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- These values are characterized but not 100% production tested.

# Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

## EXPLANATION OF THE AC SYMBOLS

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. the designations are:

A – Address

C – Clock

D – Input data

H – Logic level high

I – Instruction (program memory contents)

L – Logic level low, or ALE

P – PSEN

Q – Output data

R – RD signal

t – Time

V – Valid

W – WR signal

X – No longer a valid logic level

Z – Float

Examples:  $t_{AVLL}$  = Time for address valid to ALE low.

$t_{LLPL}$  = Time for ALE low to PSEN low.

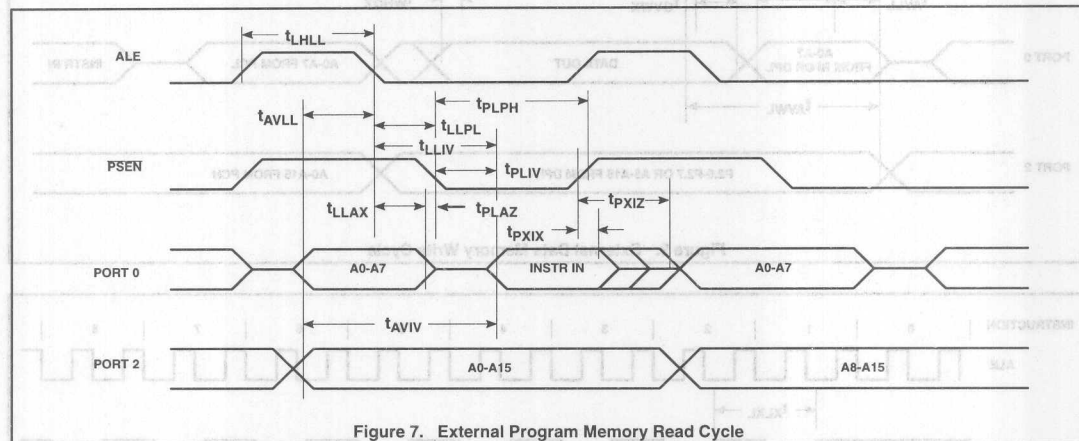


Figure 7. External Program Memory Read Cycle

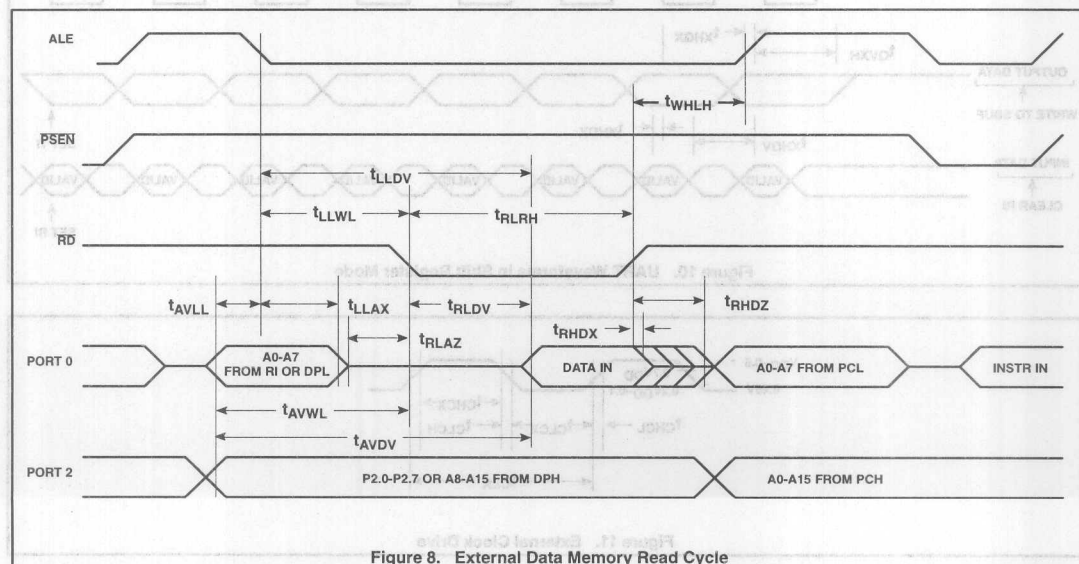
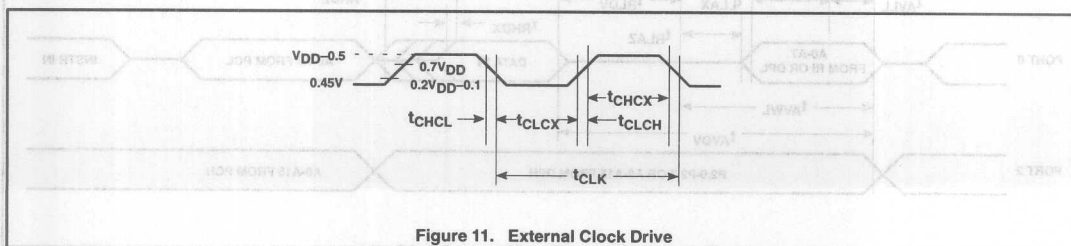
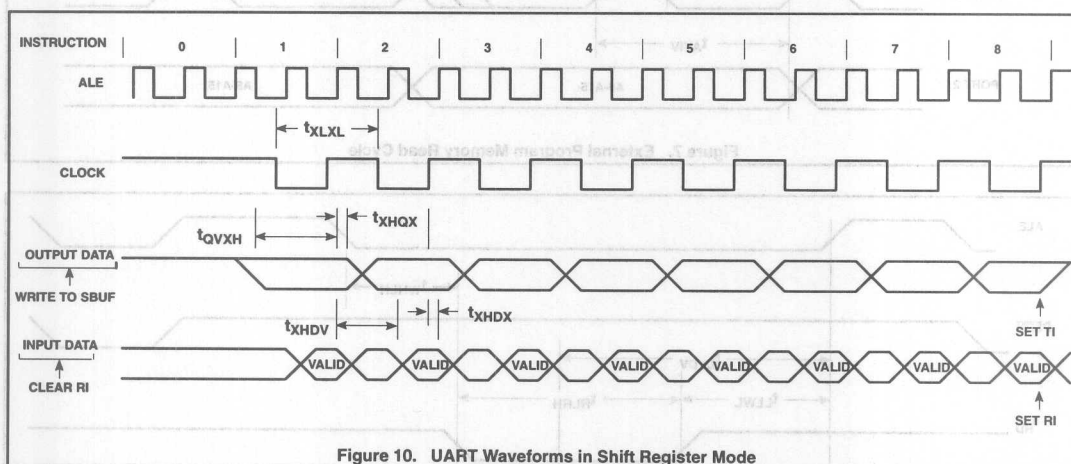
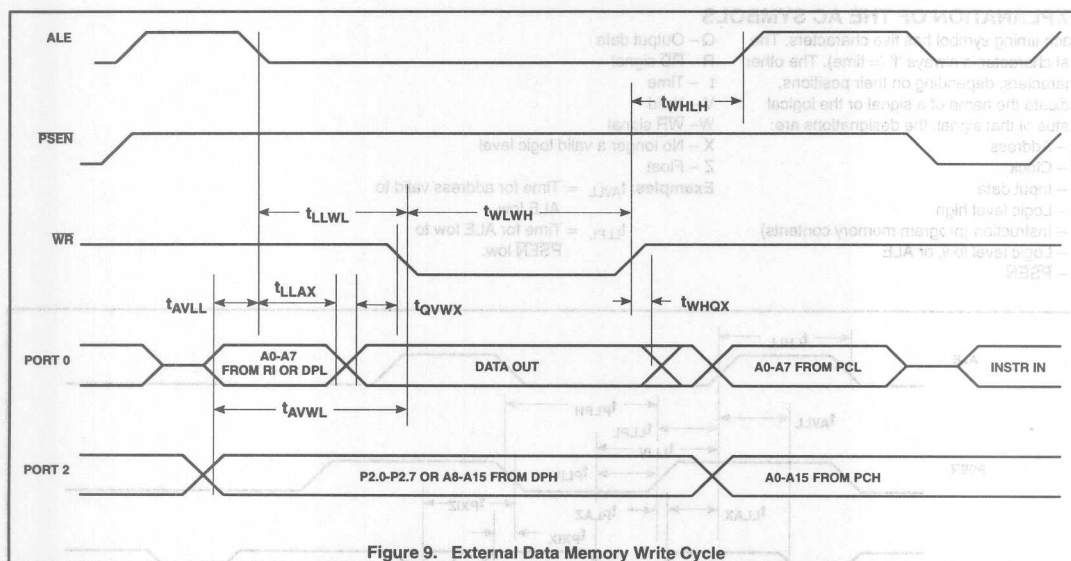


Figure 8. External Data Memory Read Cycle

# Single-chip 8-bit microcontroller with CAN controller

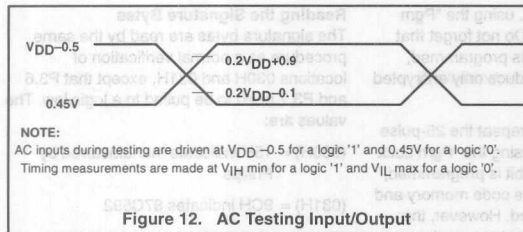
80C592/83C592/87C592



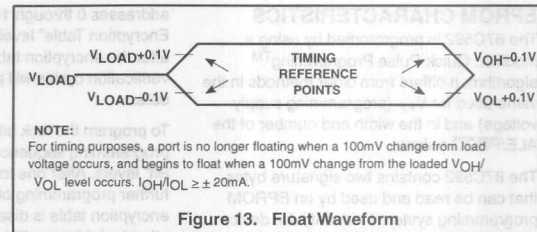


Single-chip 8-bit microcontroller  
with CAN controller

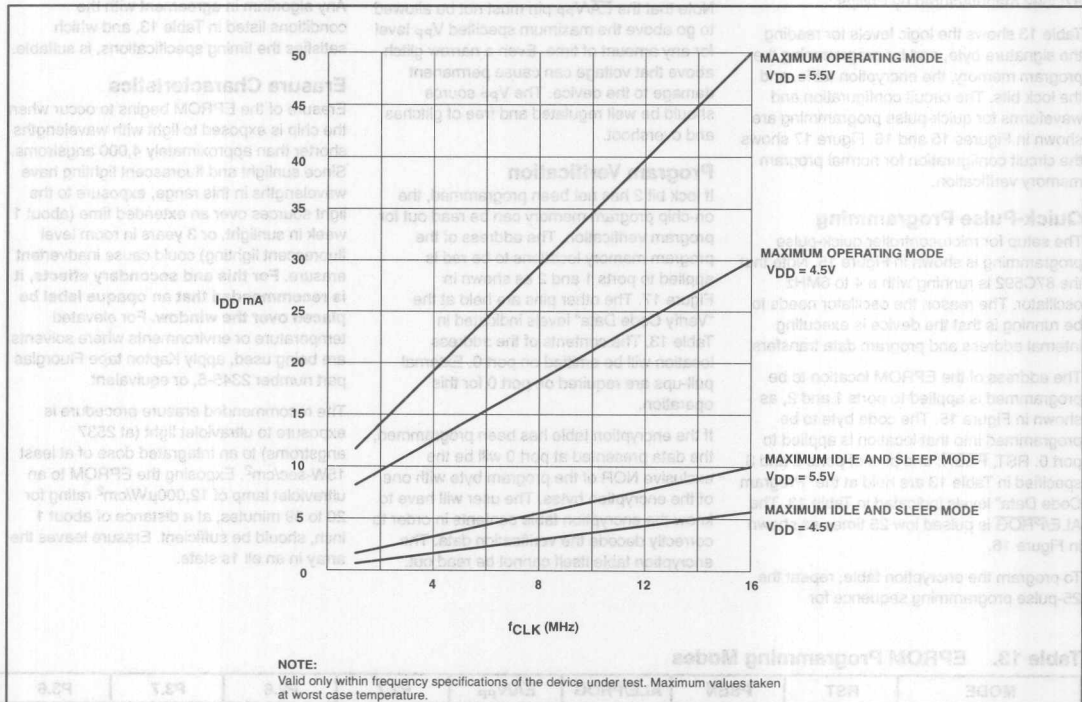
80C592/83C592/87C592



**Figure 12. AC Testing Input/Output**



**Figure 13. Float Waveform**

Figure 14.  $I_{DD}$  vs. FREQ

Valid only within frequency specifications of the device under test. Maximum values taken at worst case temperature.

**Figure 14.  $I_{DD}$  vs. FREQ**

## Single-chip 8-bit microcontroller with CAN controller

## 80C592/83C592/87C592

### EPROM CHARACTERISTICS

The 87C592 is programmed by using a modified Quick-Pulse Programming<sup>TM</sup> algorithm. It differs from older methods in the value used for  $V_{PP}$  (programming supply voltage) and in the width and number of the ALE/PROG pulses.

The 87C592 contains two signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes identify the device as an 87C592 manufactured by Philips.

Table 13 shows the logic levels for reading the signature byte, and for programming the program memory, the encryption table, and the lock bits. The circuit configuration and waveforms for quick-pulse programming are shown in Figures 15 and 16. Figure 17 shows the circuit configuration for normal program memory verification.

### Quick-Pulse Programming

The setup for microcontroller quick-pulse programming is shown in Figure 15. Note that the 87C592 is running with a 4 to 6MHz oscillator. The reason the oscillator needs to be running is that the device is executing internal address and program data transfers.

The address of the EPROM location to be programmed is applied to ports 1 and 2, as shown in Figure 15. The code byte to be programmed into that location is applied to port 0. RST, PSEN, and pins of ports 2 and 3 specified in Table 13 are held at the "Program Code Data" levels indicated in Table 13. The ALE/PROG is pulsed low 25 times as shown in Figure 16.

To program the encryption table, repeat the 25-pulse programming sequence for

addresses 0 through 1FH, using the "Pgm Encryption Table" levels. Do not forget that after the encryption table is programmed, verification cycles will produce only encrypted data.

To program the lock bits, repeat the 25-pulse programming sequence using the "Pgm Lock Bit" levels. After one lock bit is programmed, further programming of the code memory and encryption table is disabled. However, the other lock bit can still be programmed.

Note that the  $\overline{EA}/V_{PP}$  pin must not be allowed to go above the maximum specified  $V_{PP}$  level for any amount of time. Even a narrow glitch above that voltage can cause permanent damage to the device. The  $V_{PP}$  source should be well regulated and free of glitches and overshoot.

### Program Verification

If lock bit 2 has not been programmed, the on-chip program memory can be read out for program verification. The address of the program memory locations to be read is applied to ports 1 and 2 as shown in Figure 17. The other pins are held at the "Verify Code Data" levels indicated in Table 13. The contents of the address location will be emitted on port 0. External pull-ups are required on port 0 for this operation.

If the encryption table has been programmed, the data presented at port 0 will be the exclusive NOR of the program byte with one of the encryption bytes. The user will have to know the encryption table contents in order to correctly decode the verification data. The encryption table itself cannot be read out.

### Reading the Signature Bytes

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. The values are:

(030H) = 15H indicates manufactured by Philips

(031H) = 9CH indicates 87C592

### Program/Verify Algorithms

Any algorithm in agreement with the conditions listed in Table 13, and which satisfies the timing specifications, is suitable.

### Erase Characteristics

Erase of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to the light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent erasure. **For this and secondary effects, it is recommended that an opaque label be placed over the window.** For elevated temperature or environments where solvents are being used, apply Kapton tape Fluorglas part number 2345-5, or equivalent.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 angstroms) to an integrated dose of at least 15W-sec/cm<sup>2</sup>. Exposing the EPROM to an ultraviolet lamp of 12,000µW/cm<sup>2</sup> rating for 20 to 39 minutes, at a distance of about 1 inch, should be sufficient. Erasure leaves the array in an all 1s state.

Table 13. EPROM Programming Modes

MODE	RST	PSEN	ALE/PROG	$\overline{EA}/V_{PP}$	P2.7	P2.6	P3.7	P3.6
Read signature	1	0	1	1	0	0	0	0
Program code data	1	0	0*	$V_{PP}$	1	0	1	1
Verify code data	1	0	1	1	0	0	1	1
Pgm encryption table	1	0	0*	$V_{PP}$	1	0	1	0
Pgm lock bit 1	1	0	0*	$V_{PP}$	1	1	1	1
Pgm lock bit 2	1	0	0*	$V_{PP}$	1	1	0	0

#### NOTES:

1. 0 = Valid low for that pin; 1 = valid high for that pin.

2.  $V_{PP} = 12.75V \pm 0.25V$ .

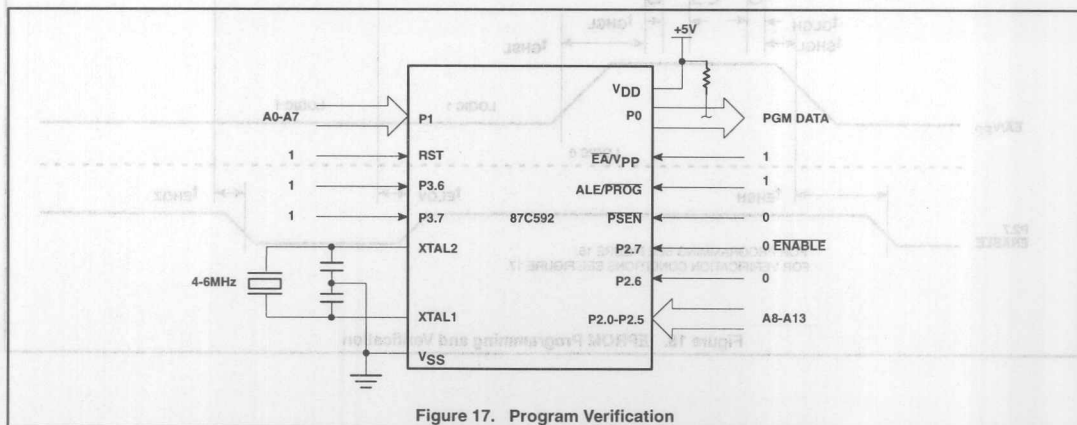
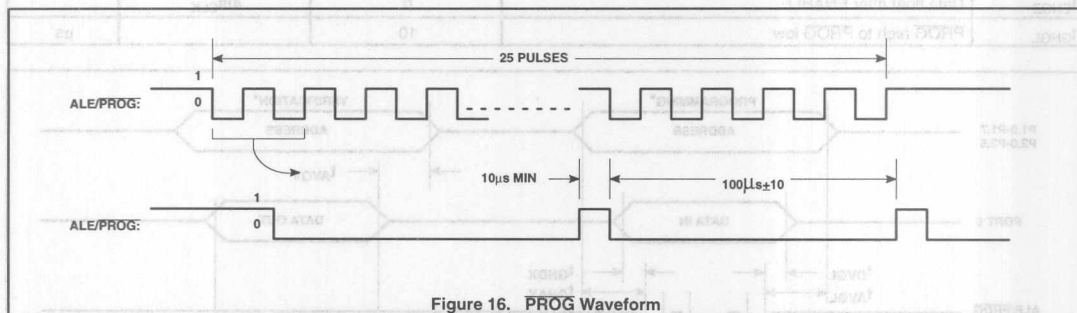
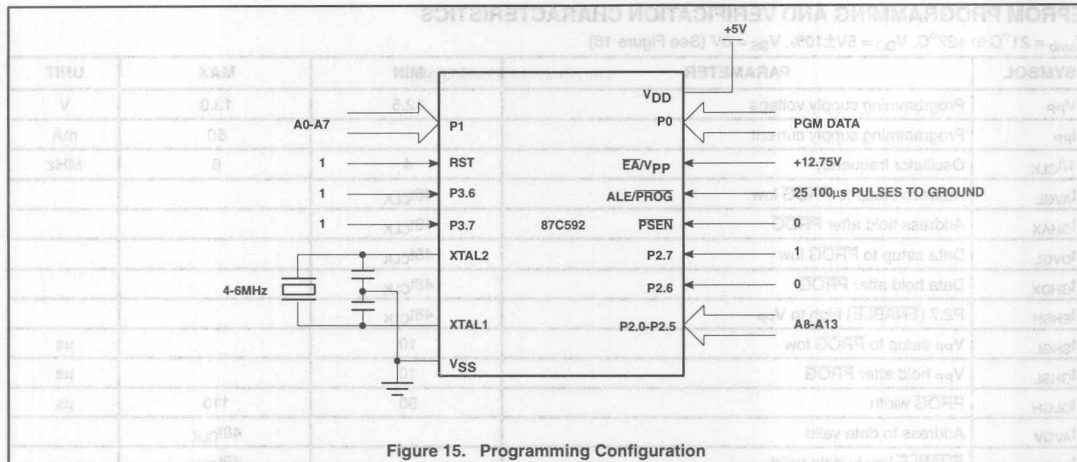
3.  $V_{DD} = 5V \pm 10\%$  during programming and verification.

\* ALE/PROG receives 25 programming pulses while  $V_{PP}$  is held at 12.75V. Each programming pulse is low for 100µs ( $\pm 10\mu s$ ) and high for a minimum of 10µs.

<sup>TM</sup>Trademark phrase of Intel Corporation.

Single-chip 8-bit microcontroller  
with CAN controller

80C592/83C592/87C592



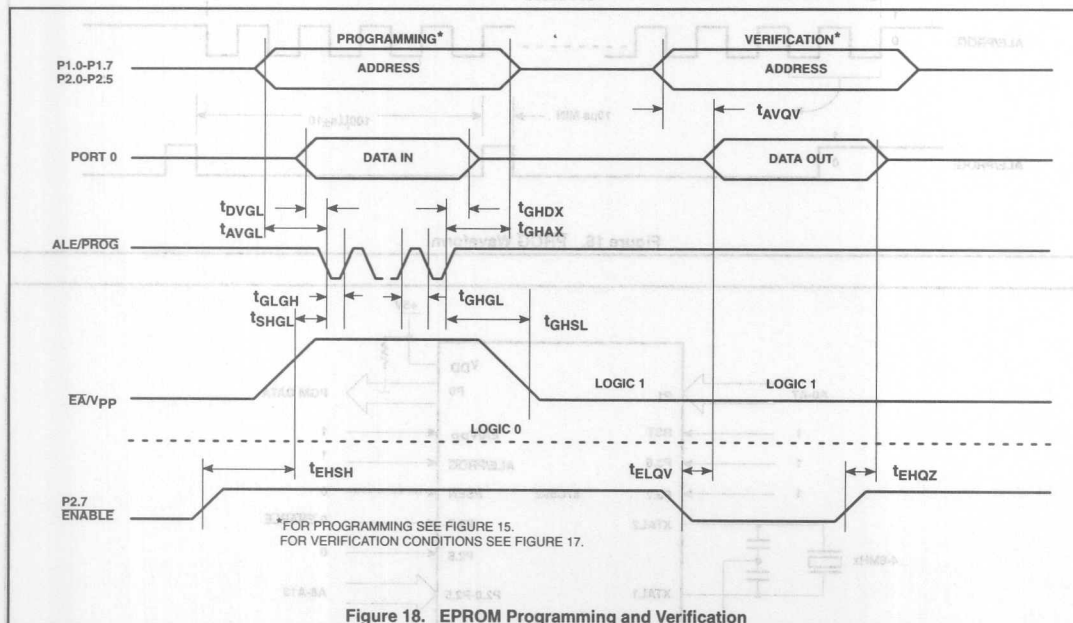
# Single-chip 8-bit microcontroller with CAN controller

80C592/83C592/87C592

## EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

 $T_{amb} = 21^{\circ}\text{C}$  to  $+27^{\circ}\text{C}$ ,  $V_{DD} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$  (See Figure 18)

SYMBOL	PARAMETER	MIN	MAX	UNIT
$V_{PP}$	Programming supply voltage	12.5	13.0	V
$I_{PP}$	Programming supply current		50	mA
$1/t_{CLK}$	Oscillator frequency	4	6	MHz
$t_{AVGL}$	Address setup to PROG low	48tCLK		
$t_{GHAX}$	Address hold after PROG	48tCLK		
$t_{DVGL}$	Data setup to PROG low	48tCLK		
$t_{GHDX}$	Data hold after PROG	48tCLK		
$t_{EHS}$	P2.7 (ENABLE) high to $V_{PP}$	48tCLK		
$t_{SHGL}$	$V_{PP}$ setup to PROG low	10		$\mu\text{s}$
$t_{GHSL}$	$V_{PP}$ hold after PROG	10		$\mu\text{s}$
$t_{GLGH}$	PROG width	90	110	$\mu\text{s}$
$t_{AVQV}$	Address to data valid		48tCLK	
$t_{ELQV}$	ENABLE low to data valid		48tCLK	
$t_{EHQZ}$	Data float after ENABLE	0	48tCLK	
$t_{GHGL}$	PROG high to PROG low	10		$\mu\text{s}$



## 8-bit microcontroller with on-chip CAN

## P8xCE598

## 1.0 FEATURES

- 80C51 central processing unit (CPU)
- 32 kbyte on-chip ROM (EPROM), externally expandable to 64 kbyte
- 2 × 256 byte on-chip RAM, externally expandable to 64 kbyte
- Two standard 16-bit timer/counters
- One additional 16-bit timer/counter coupled to four capture and three compare registers
- A 10-bit ADC with eight multiplexed analog inputs
- Two 8-bit resolution, pulse width modulation outputs
- 15 interrupt sources with 2 priority levels (2 to 6 external interrupt sources possible)
- Five 8-bit I/O ports plus one 8-bit input port shared with analog inputs
- CAN controller with DMA transfer facility to internal RAM
- 1 Mbit/s CAN-Controller with bus failure management facility
- $V_{DD}/2$  reference voltage
- Full-duplex UART compatible with the standard 80C51
- On-chip watchdog timer
- 1.2 to 16MHz clock frequency (3.5 to 16MHz for EPROM/OTP version)
- Electro-Magnetic Compatibility (EMC) improvements

## 2.0 GENERAL DESCRIPTION

The P80CE598/P83CE598/P87CE598 (hereafter generically referred to as P8xCE598) is a single-chip 8-bit high-performance microcontroller with on-chip CAN-controller designed for use in automotive and general industrial applications. In addition to the 80C51 standard features, this device provides a number of dedicated hardware functions highly suitable for automotive and general industrial applications. Figure 1 illustrates the block diagram of the P8xCE598.

Three versions of the P8xCE598 will be offered:

- P83CE598 (ROM version)
- P80CE598 (ROMless version)
- P87CE598 (EPROM/OTP version) (OTP = One Time Programming)

The P8xCE598 basically combines the functions of the existing P8xC552 and the Philips CAN-controller PCA82C200 (CAN =

Controller Area Network) with the following enhanced features:

- 32 kbyte Program Memory
- 2 × 256 byte Data Memory
- DMA between CAN Transmit/Receive Buffer and internal RAM

The temperature range includes a -40 to +85°C version for general applications and an automotive temperature range version of -40 to +125°C for the ROM and ROMless versions with a maximum clock frequency of 16MHz. The P87CE598 (EPROM/OTP version) has a temperature range of -40 to +85°C.

The main differences to the P8xC552 microcontroller are:

- A CAN-controller substitutes the I<sup>2</sup>C-serial interface
- 32 kbyte programmable ROM respectively, EPROM (P8xC552 has 8 kbyte)
- Additional 256 byte RAM

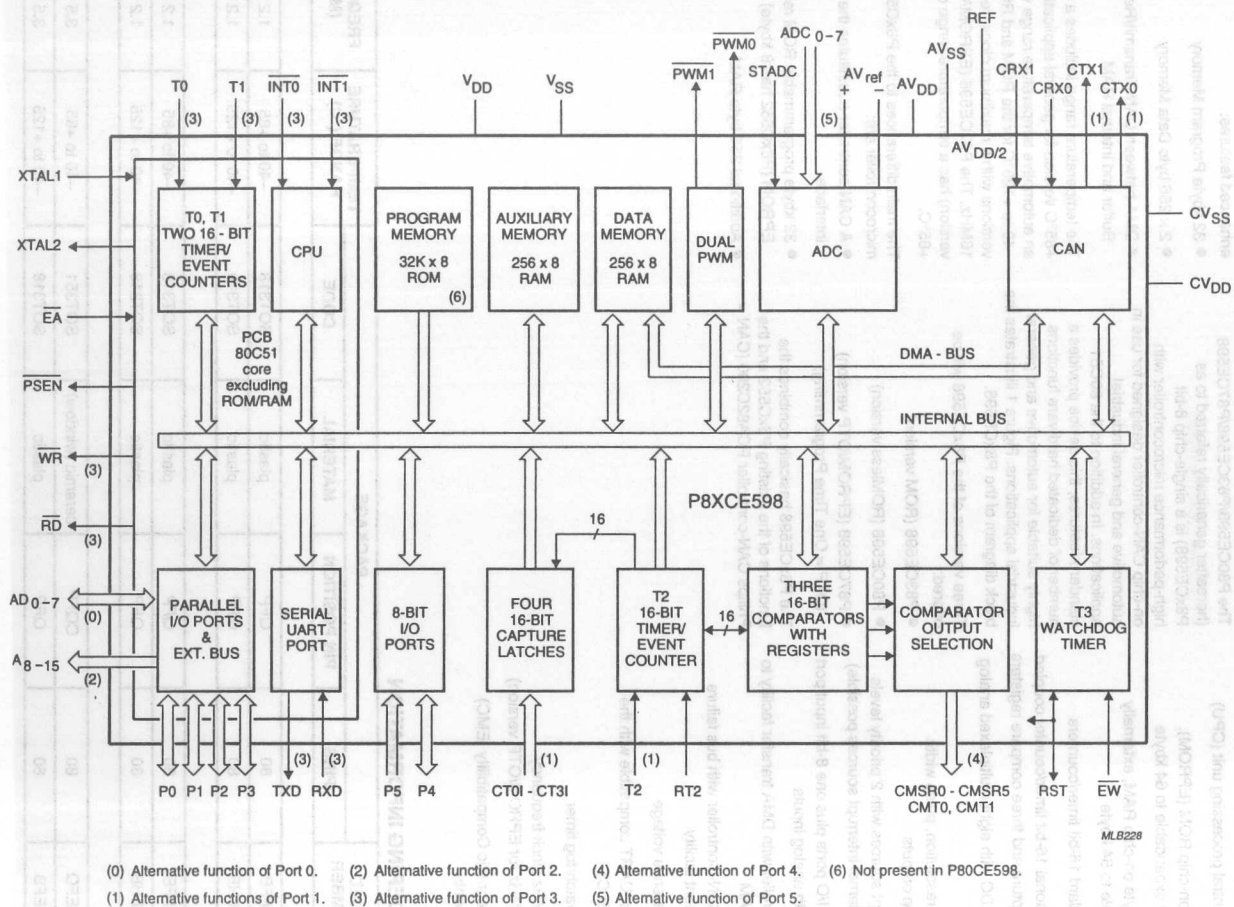
## 3.0 ORDERING INFORMATION

EXTENDED TYPE NUMBER	PACKAGE				TEMPERATURE RANGE (°C)	FREQUENCY (MHz)
	PINS	PIN POSITION	MATERIAL	CODE		
ROMless						
P80CE598FFB	80	QFP	plastic	SOT318	−40 to +85	1.2 to 16
P80CE598FHB	80	QFP	plastic	SOT318	−40 to +125	1.2 to 16
ROM						
P83CE598FFB	80	QFP	plastic	SOT318	−40 to +85	1.2 to 16
P83CE598FHB	80	QFP	plastic	SOT318	−40 to +125	1.2 to 16
EPROM						
P87CE598EFQ	80	CQFP	ceramic (window)	SOT351	−40 to +85	3.5 to 16
P87CE598EFB	80	QFP	plastic	SOT318	−40 to +125	3.5 to 16



# 8-bit microcontroller with on-chip CAN

P8XCE598



## 8-bit microcontroller with on-chip CAN P8xCE598

### 4.0 ELECTROMAGNETIC COMPATIBILITY (EMC)

Primary attention is paid to the reduction of electro-magnetic emission of the microcontroller P8xCE598. The following features reduce the electro-magnetic emission and additionally improve the electromagnetic susceptibility:

- One analog supply voltage pin ( $AV_{DD}$ ) and one analog ground pin ( $AV_{SS}$ ) as a pair of pins placed on one side of the package (see Figure ) to provide:
  - analog power supply (+5V) and analog ground for ADC, CAN receiver and reference voltage.
- Four digital supply voltage pins ( $V_{DD1;2;3;4}$ ) and four digital ground pins ( $V_{SS1;2;3;4}$ ) are provided on the package. From these pins, one  $V_{DD}$  and one  $V_{SS}$  as a pair of pins are placed on each of the four sides of the package to provide:
  - $V_{DD1}/V_{SS1}$  for internal logic (CPU, Timers/counters, Memory, CAN, UART, ADC)
  - $V_{DD2}/V_{SS2}$  for Port 1, Port 3 and Port 4, and  $PWM0$  and  $PWM1$  outputs
  - $V_{DD3}/V_{SS3}$  for the on-chip oscillator
- $V_{DD4}/V_{SS4}$  for the Port 0, Port 2, ALE output and PSEN output
- One CAN supply voltage pin ( $CV_{DD}$ ) and one CAN ground pin ( $CV_{SS}$ ) as a pair of pins placed on one side of the package to provide:
  - Power supply (+5V) and ground for the CAN transmitter outputs
- One the P87CE598 version, the additional supply voltage pin ( $V_{PP}$ ) to provide:
  - the programming supply voltage  $V_{PP}$ , the alternative function for pin 65 ( $EA/V_{PP}$ ) on the P87CE598 EPROM/OTP version (see Figure 13).
- Internal decoupling capacitance improves the EMC radiation behaviour and the EMC immunity.
- External capacitors are to be located as close as possible between pins  $V_{DDX}$  and  $V_{SSX}$  cross-connected in the following method (i.e.,  $V_{DD1}$  to  $V_{SS2}$ , etc.); ceramic chip capacitors are recommended (100nF).

of  $f_{OSC}/6$ ) can be disabled under software control (bit 5 in the PCON SFR: "RFI"); if disabled, no ALE pulse will occur. ALE pin will be pulled down internally, switching an external address latch to a quiet state. The MOVX instruction will still toggle ALE as a normal MOVX. ALE will retain its normal HIGH value during idle mode and a LOW value during power-down mode while in the "RFI" reduction mode.

Additionally during internal access ( $E\bar{A} = 1$ ) ALE will toggle normally when the address exceeds the internal program memory size. During external access ( $E\bar{A} = 0$ ) ALE will always toggle normally, whether the flag "RFI" is set or not.

### 5.0 OTHER FEATURES

The 8xCE598 has the same operation as the 8XC552 for the A/D, Timers, PWM outputs, UART and Watchdog Timer. Please refer to the 8XC552 datasheet for details on these features. The CAN interface and AUX RAM on the 8xCE598 are discussed in detail in the 8XC592 datasheet. Please refer to the 8XC592 datasheet for information on these features.

### 4.1 Recommendation on ALE

For applications that require no external memory or temporarily no external memory: the ALE output signal (pulses at a frequency

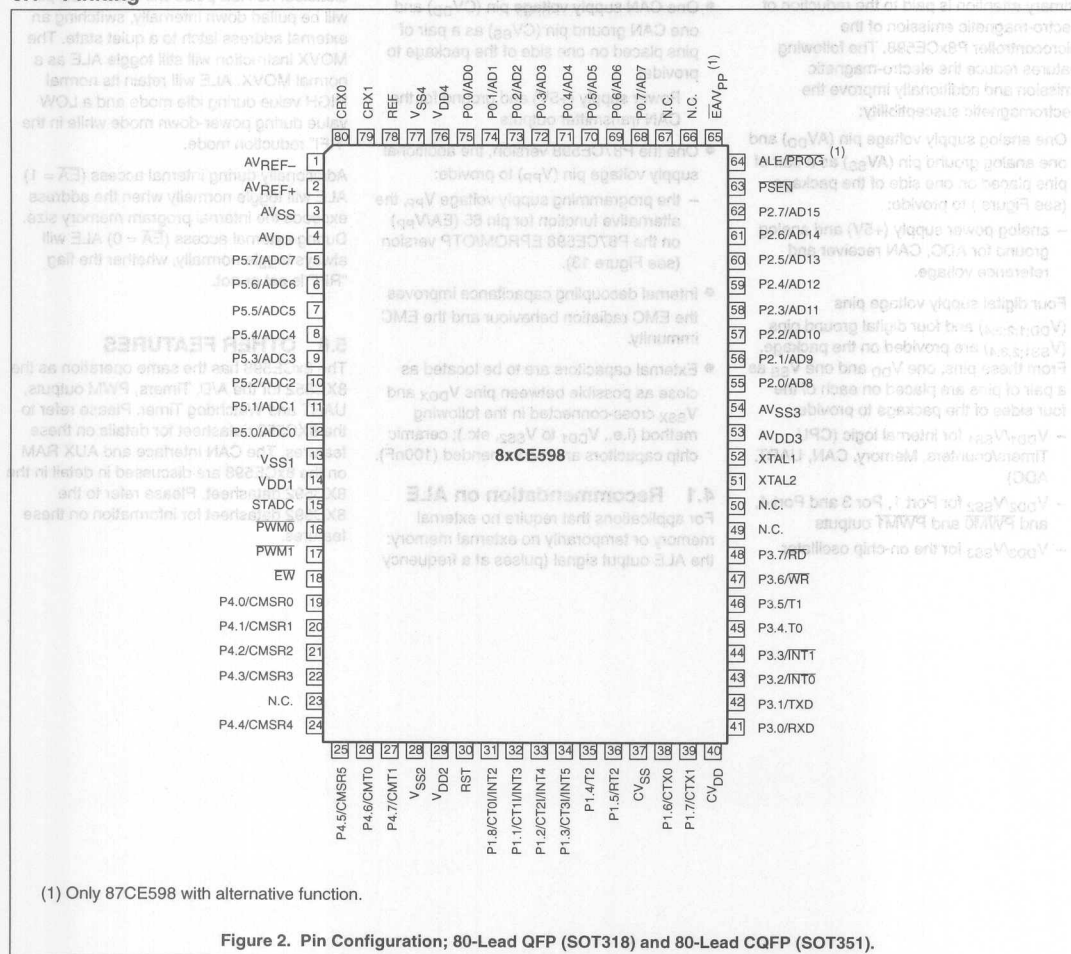
Figure 2. Pin Configuration: 60-Lead QFP (80T318) and 80-Lead CQFP (80T321)

## 8-bit microcontroller with on-chip CAN

## P8xCE598

## 6.0 PINNING INFORMATION

## 6.1 Pinning



## 8-bit microcontroller with on-chip CAN P8xCE598

## 6.2 Pin Description

SYMBOL	PIN NO.	DESCRIPTION	DESCRIPTION	PIN NO.	SYMBOL
AV <sub>REF-</sub>	1	<b>Low-end of ADC</b> (analog to digital conversion) reference resistor.		41	P5.7
AV <sub>REF+</sub>	2	<b>High-end of ADC</b> (analog to digital conversion) reference resistor. The level of AV <sub>REF+</sub> must be higher than that of AV <sub>REF-</sub> .		42	P5.6
AV <sub>SS</sub>	3	<b>Analog ground</b> for ADC, CAN receiver and reference voltage.		43	P5.5
AV <sub>DD</sub>	4	<b>Analog power supply</b> (+5V) for ADC, CAN receiver and reference voltage.		44	P5.4
P5.7 – P5.0	5 – 12	<b>Port 5: 8-bit input port.</b>		45	P5.3
P5.7 – P5.0/ ADC7 – ADC0	5 – 12	<b>Port 5 alternative functions:</b> 8 input channels to ADC		46	P5.2
V <sub>SS1</sub>	13	<b>Digital ground:</b> for internal logic (CPU, Timers/counters, Memory, CAN, UART, ADC).		47	P5.1
V <sub>DD1</sub>	14	<b>Digital power supply:</b> for internal logic (CPU Timers/counters, Memory, CAN, UART, ADC).		48	P5.0
STADC	15	<b>Start ADC operation:</b> input starting analog to digital conversion triggered by a rising edge (ADC operation can also be started by software). This pin must not float.		49	P5.7
PWM0	16	<b>Pulse Width Modulation</b> output 0.		50	P5.6
PWM1	17	<b>Pulse Width Modulation</b> output 1.		51	P5.5
EW	18	<b>Enable Watchdog Timer (WDT):</b> enable for T3 Watchdog Timer and disable Power-down mode. this pin must not float.		52	P5.4
P4.0 – P4.7	19 – 22, 24 – 27	<b>Port 4: 8-bit quasi-bidirectional I/O port.</b>		53	P5.3
P4.0/CMSR0	19	<b>Port 4 alternative functions:</b> Timer T2: Compare and set/reset outputs on a match with timer T2.		54	P5.2
P4.1/CMSR1	20	Timer T2: Compare and set/reset outputs on a match with timer T2.		55	P5.1
P4.2/CMSR2	21	Timer T2: Compare and set/reset outputs on a match with timer T2.		56	P5.0
P4.3/CMSR3	22	Timer T2: Compare and set/reset outputs on a match with timer T2.		57	P5.7
P4.4/CMSR4	24	Timer T2: Compare and set/reset outputs on a match with timer T2.		58	P5.6
P4.5/CMSR5	25	Timer T2: Compare and set/reset outputs on a match with timer T2.		59	P5.5
P4.6/CMT0	26	Timer T2: Compare and toggle outputs on a match with timer T2.		60	P5.4
P4.7/CMT1	27	Timer T2: Compare and toggle outputs on a match with timer T2.		61	P5.3
V <sub>DD2</sub>	28	<b>Digital power supply:</b> for Port 1, Port 3 and Port 4, and PWM0 and PWM1 outputs.		62	P5.2
V <sub>SS2</sub>	29	<b>Digital ground:</b> for Port 1, Port 3 and Port 4, and PWM0 and PWM1 outputs.		63	P5.1
RST	30	<b>Reset:</b> Input to reset the P8xCE598. It also provides a reset pulse as output when timer T3 overflows or after a CAN wake-up from Power-down.		64	P5.0
P1.0 – P1.7	31 – 36, 38 – 39	<b>Port 1: 8-bit quasi-bidirectional I/O port.</b>		65	P5.7
P1.0/CT0I/INT2	31	<b>Port 1 alternative functions:</b> Capture timer inputs for Timer T2 or external interrupt inputs.		66	P5.6
P1.1/CT1I/INT3	32	Capture timer inputs for Timer T2 or external interrupt inputs.		67	P5.5
P1.2/CT2I/INT4	33	Capture timer inputs for Timer T2 or external interrupt inputs.		68	P5.4
P1.3/CT3I/INT5	34	Capture timer inputs for Timer T2 or external interrupt inputs.		69	P5.3
P1.4/T2	35	T2 event input, rising edge triggered.		70	P5.2
P1.5/RT2	36	T2 timer reset input, rising edge triggered.		71	P5.1
P1.6/CTX0	38	CAN transmitter output 0 (note 1).		72	P5.0
P1.7/CTX1	39	CAN transmitter output 1 (note 1).		73	P5.7
CV <sub>SS</sub>	37	<b>Ground</b> potential for the CAN transmitter outputs.		74	P5.6
CV <sub>DD</sub>	40	<b>Power supply</b> (+5V) for the CAN transmitter outputs.		75	P5.5

- NOTES:
1. If the CAN controller is in the reset state (e.g., after a power-up reset), CAN Control Register bit 0, the CAN transmitter outputs are floating and the pins P1.6 and P1.7 can be used as open-drain pins. After a power-up reset the port data is HIGH, leaving the pins P1.6 and P1.7 floating.
  2. For P8xCE598 microcontroller specified with the option "ROM-code protection", the EA pin is tied to ground during reset and is "don't care" after reset. Release of whether the ROM-code protection is selected or not.
  3. After an external reset (RST = HIGH) the internal AV<sub>DD2</sub> source is activated and the pin REF is a reference output. During Power-down mode the AV<sub>DD2</sub> source is switched off if the CAN controller is in the reset state (e.g., after an external reset).
  4. To avoid a "kick up" effect at power-on, the voltage on any pin (at any time) must not be higher or lower than V<sub>DD</sub>+0.5V or V<sub>SS</sub>-0.5V, respectively.

# 8-bit microcontroller with on-chip CAN P8xCE598

## 6.2 Pin Description (Continued)

SYMBOL	PIN NO.	DESCRIPTION
P3.0 – P3.7	41 – 48	<b>Port 3:</b> 8-bit quasi-directional I/O port. <b>Port 3 alternative functions:</b>
P3.0/RXD	41	Serial Input Port.
P3.1/TXD	42	Serial Output Port.
P3.2/INT0	43	External interrupt.
P3.3/INT1	44	External interrupt.
P3.4/T0	45	Timer 0 external input.
P3.5/T1	46	Timer 1 external input.
P3.6/WR	47	External data memory <b>Write Strobe</b> .
P3.7/RD	48	External data memory <b>Read Strobe</b> .
XTAL2	51	<b>Crystal pin 2:</b> output of the inverting amplifier that forms the oscillator. Left open-circuit when and external oscillator clock is used.
XTAL1	52	<b>Crystal pin 1:</b> input of the inverting amplifier that forms the oscillator, and input to the internal clock generator. Receives the external oscillator clock signal when an external oscillator is used.
V <sub>DD3</sub>	53	<b>Digital power supply:</b> for the on-chip oscillator.
V <sub>SS3</sub>	54	<b>Digital ground:</b> for the on-chip oscillator.
P2.0 – P2.7	55 – 62	<b>Port 2:</b> 8-bit quasi-directional I/O port. <b>Port 2 alternative functions:</b>
P2.0 – P2.7/ A08 – A15	55 – 62	<b>High-order address byte</b> for external memory (A08 – A15). Port 2 can sink/source 1 TTL (= 4 LSTTL) input.
PSEN	63	<b>Program Store Enable:</b> active LOW read strobe to external program memory. PSEN can drive 8 LSTTL inputs.
ALE/PROG	64	<b>Address Latch Enable:</b> latches the low-byte of the address during accesses to external memory. It is activated every six oscillator periods. During an external data memory access one ALE pulse is skipped. ALE can drive up to 8 LSTTL inputs and handles CMOS inputs without an external pull-up. <b>Pin 64 alternative function:</b> The alternative function for pin 64 on the P87CE598 is the programming-pulse input PROG.
EA/V <sub>PP</sub>	65	<b>External Access:</b> When EA is held at TTL level HIGH during reset, the CPU executes out of the internal Program Memory provided the program counter is less than 16384. When EA is held at LOW level during reset, the CPU executes out of the external Program Memory. The EA pin is latched during reset and is "don't care" after reset (note 2). <b>Pin 65 alternative function:</b> The alternative function for pin 65 on the P87CE598 is the programming supply voltage V <sub>PP</sub> .
P0.7 – P0.0	68 – 75	<b>Port 0:</b> 8-bit open drain bidirectional I/O port. <b>Port 0 alternative functions:</b>
P0.7 – P0.0/ AD7 – AD0	68 – 75	<b>Multiplexed low-order address</b> and data bus for external memory (AD0 – AD7). Port 0 can sink/source up to 8 LSTTL inputs.
V <sub>DD4</sub>	76	<b>Digital power supply:</b> for the Port 0, Port 2, ALE output and PSEN output.
V <sub>SS4</sub>	77	<b>Digital ground:</b> for the Port 0, Port 2, ALE output and PSEN output.
REF	78	<b>AV<sub>DD</sub>/2 reference voltage</b> output respectively, input depending on CAN Control Register bit 5. If the internal reference is used, then REF should be connected to AV <sub>SS</sub> via a 10nF (or more) capacitor (note 3).
CRX1 and CRX0	79 and 80	<b>Inputs from the CAN</b> bus line to the differential input comparator of the on-chip CAN-controller. A CRX0 level higher than that of CRX1 is interpreted as a logical "1" (recessive) and vice versa for a logical "0" (dominant).
N.C.	23, 49, 50, 66, 67	No Connection (both SOT318 and SOT351).

### NOTES:

- If the CAN-controller is in the reset state (e.g., after a power-up reset; CAN Control Register bit 0), the CAN transmitter outputs are floating and the pins P1.6 and P1.7 can be used as open-drain port pins. After a power-up reset the port data is HIGH, leaving the pins P1.6 and P1.7 floating.
- For P83CXXX microcontrollers specified with the option "ROM-code protection", the EA pin is latched during reset and is "don't care" after reset, regardless of whether the ROM-code protection is selected or not.
- After an external reset (RST = HIGH) the internal AV<sub>DD</sub>/2 source is activated and the pin REF is a reference output. During Power-down mode the AV<sub>DD</sub>/2 source is switched off if the CAN-controller is in the reset state (e.g., after an external reset).
- To avoid a "latch up" effect at power-on, the voltage on any pin (at any time) must not be higher or lower than V<sub>DD</sub>+0.5V or V<sub>SS</sub>-0.5V respectively.



## 8-bit microcontroller with on-chip CAN

## P8xCE598

## 7.0 LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134).

SYMBOL	PARAMETER	MIN.	MAX.	UNIT
$V_{DD}$	Voltage on $V_{DD}$ pin	-0.5	+6.5	V
$V_I$	Input voltage on any pin (except from CTX0, CTX1, CRX0 and CRX1)	-0.5	$V_{DD}+0.5$	V
$\pm I_{I, O}$	Input/output current on any single I/O pin (except from CTX0 and CTX1)		10	mA
$I_{OT}$	Sink current of CTX0, CTX1 together	-	30	mA
$-I_{OT}$	Source current of CTX0, CTX1 together		20	mA
$P_{tot}$	Total power dissipation (note 1)		1.0	W
$T_{stg}$	Storage temperature range	-65	+150	°C
$T_{amb}$	Operating ambient temperature range: P87CE598EFQ/EFB P83CE598FFB P83CE598FHB	-40 -40 -40	+85 +85 +125	°C

## NOTE:

1. This value is based on the maximum allowable die temperature and the thermal resistance of the package, no on device power consumption.

## 8.0 DC CHARACTERISTICS

$V_{DD} = 5V$  ( $\pm 10\%$ );  $V_{SS} = 0V$ ;  $T_{amb} = -40$  to  $+125^\circ C$  for the P83CE598FHB and  $T_{amb} = -40$  to  $+85^\circ C$  for the P83CE598FFB and P87CE598EFx; all voltages with respect to  $V_{SS}$  unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
<b>Supply</b>					
$V_{DD}$	Supply voltage range		4.5	5.5	V
$I_{DD}$	Supply current:				
$I_{ID}$	Operating	$f_{CLK} = 16MHz$ ; note 1	-	50	mA
$I_{IS}$	Idle mode	$f_{CLK} = 16MHz$ ; note 2	-	15	mA
$I_{PD}$	Idle and sleep mode	$f_{CLK} = 16MHz$ ; note 3	-	10	mA
	Power-down mode:				
	P83CE598FHB		-	150	$\mu A$
	P83CE598FFB		-	100	$\mu A$
	P87CE598EFB/Q		-	100	$\mu A$
<b>Inputs</b>					
$V_{IL}$	LOW level input voltage (except EA, CRX0, CRX1)		-0.5	$0.2V_{DD}-0.1$	V
$V_{IL1}$	LOW level input voltage EA		-0.5	$0.2V_{DD}-0.3$	V
$V_{IH}$	HIGH level input voltage (except RST, XTAL1, CRX0, CRX1)		$0.2V_{DD}+0.9$	$V_{DD}+0.5$	V
$V_{IH1}$	HIGH level input voltage RST and XTAL1		$0.7V_{DD}$	$V_{DD}+0.5$	V
$I_{IL}$	LOW level input current Ports 1, 2, 3 and 4	$V_I = 0.45V$	-	-50	$\mu A$
$I_{TL}$	Input current HIGH-to-LOW transition Ports 1, 2, 3 and 4 (except P1.6 and P1.7)	$V_I = 2.0V/0.45V$	-	-650	$\mu A$
$I_{LI1}$	Input leakage current Port 0, EA, STADC, EW, P1.6, P1.7	$0.45V < V_I < V_{DD}$	-	$\pm 10$	$\mu A$
$I_{LI2}$	Input leakage current Port 5	$0.45V < V_I < V_{DD}$	-	$\pm 1$	$\mu A$

## 8-bit microcontroller with on-chip CAN

## P8xCE598

## DC CHARACTERISTICS (Continued)

$V_{DD} = 5V$  ( $\pm 10\%$ );  $V_{SS} = 0V$ ;  $T_{amb} = -40$  to  $+125^\circ C$  for the P83CE598FHB and  $T_{amb} = -40$  to  $+85^\circ C$  for the P83CE598FFB and P87CE598EFx; all voltages with respect to  $V_{SS}$  unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
<b>Outputs</b>					
$I_{OL}$	LOW level input current Ports 1, 2, 3 and 4	$I_{OL} = 1.6mA$ notes 5 and 6	—	0.45	V
$V_{OL1}$	LOW level output voltage Port 0, ALE, PSEN, PWM0, PWM1, P1.6, P1.7	$I_{OL} = 3.2mA$ notes 5 and 6	—	0.45	V
$V_{OH}$	HIGH level output voltage Ports 1, 2, 3 and 4 (except P1.6 and P1.7)	$I_{OH} = -60\mu A$ $I_{OH} = -25\mu A$ $I_{OH} = -10\mu A$	2.4 0.75 $V_{DD}$ 0.9 $V_{DD}$	— — —	V V V
$V_{OH1}$	HIGH level output voltage Port 0 in external bus mode, ALE, PSEN, PWM0, PWM1	$I_{OH} = -400\mu A$ $I_{OH} = -150\mu A$ $I_{OH} = -40\mu A$ note 7	2.4 0.75 $V_{DD}$ 0.9 $V_{DD}$	— — —	V V V
$V_{OH2}$	HIGH level output voltage RST	$I_{OH} = -400\mu A$ $I_{OH} = -120\mu A$	2.4 0.8 $V_{DD}$	— —	V V
$R_{RST}$	RST pull-down resistor		50	150	k $\Omega$
$C_{I/O}$	I/O pin capacitance	test freq. = 1MHz; $T_{amb} = 25^\circ C$	—	10	pF
<b>Analog Supply</b>					
$AV_{DD}$	Analog supply voltage range	$AV_{DD} = V_{DD} \pm 0.2V$	4.5	5.5	V
	Analog supply current:				
$AI_{DD}$	Operating	Port 5 = $AV_{DD}$	—	2.5	mA
$AI_{ID}$	Idle mode		—	2.5	mA
$AI_{IS}$	Idle and sleep mode:				
	P83CE598FHB		—	400	$\mu A$
	P8xCE598xFx		—	350	$\mu A$
$AI_{PD}$	Power-down mode:				
	P83CE598FHB		—	400	$\mu A$
	P8xCE598xFx		—	350	$\mu A$
<b>Analog Inputs</b>					
$AV_{IN}$	Analog input voltage		$AV_{SS} - 0.2$	$AV_{DD} + 0.2$	V
$AV_{REF-}$	Reference voltage		$AV_{SS} - 0.2$	—	V
$AV_{REF+}$			—	$AV_{DD} + 0.2$	V
$R_{REF}$	REsistance between $AV_{REF+}$ and $AV_{REF-}$		10	50	k $\Omega$
$C_{IA}$	Analog input capacitance		—	15	pF
$t_{ADS}$	Sampling time	note 8	—	8 $t_{CY}$	$\mu s$
$t_{ADC}$	Conversion time (including sample time)	note 8	—	50 $t_{CY}$	$\mu s$
$DL_e$	Differential non-linearity	notes 9, 10, and 11	—	$\pm 1$	LSB
$IL_e$	Integral non-linearity	notes 9 and 12	—	$\pm 2$	LSB
$OS_e$	Offset error	notes 9 and 13	—	$\pm 2$	LSB
$G_e$	Gain error	notes 9 and 14	—	$\pm 0.4$	%
$A_e$	Absolute voltage error	notes 9 and 15	—	$\pm 3$	LSB
$M_{cIG}$	Channel to channel matching		—	$\pm 1$	LSB
$C_t$	Crosstalk between P5 inputs	0 to 100kHz	—	-60	dB

## 8-bit microcontroller with on-chip CAN

## P8xCE598

## DC CHARACTERISTICS (Continued)

$V_{DD} = 5V (\pm 10\%)$ ;  $V_{SS} = 0V$ ;  $T_{amb} = -40$  to  $+125^\circ C$  for the P83CE598FHB and  $T_{amb} = -40$  to  $+85^\circ C$  for the P83CE598FFB and P87CE598EFx; all voltages with respect to  $V_{SS}$  unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
<b>CAN Input Comparator (CRX0, CRX1)</b>					
$V_{DIF}$	Differential input voltage	$AV_{DD} = 5V \pm 5\%$ ; $1.4V < V_I < AV_{DD} - 1.4V$ note 16	$\pm 32$	—	mV
$V_{HYST}$	Hysteresis voltage	note 16	8	30	mV
$I_I$	Input current		—	$\pm 400$	nA
<b>CAN Output Driver</b>					
$V_{OLT}$	CTX0, CTX1, output voltage LOW	$V_{DD} = 5V \pm 5\%$ $I_O = 1.2mA$ ; note 16 $I_O = 10mA$	—	0.1 0.5	V V
$V_{OHT}$	CTX0, CTX1, output voltage HIGH	$I_O = -1.2mA$ ; note 16 $I_O = -10mA$ ; note 17	$V_{DD} - 0.1$ $V_{DD} - 0.5$	—	V V
<b>Reference</b>					
$V_{REFOUT}$	REF output voltage (bit Reference Active = HIGH)	$AV_{DD} = 5V \pm 5\%$ $-0.1mA < I_L < 0.1mA$ ; $C_L = 10nF$ ; note 16	$AV_{DD}/2 - 0.05$	$AV_{DD}/2 - 0.05$	V
$I_{REFIN}$	REF input current (bit Reference Active = LOW)	$1.5V < V_{REFIN} < AV_{DD} - 1.5V$	—	$\pm 10$	$\mu A$

## NOTES TO DC CHARACTERISTICS:

- The operating current is measured with all output pins disconnected; XTAL1 is driven with  $t_r = t_f = 10ns$ ;  $V_{IL} = V_{SS} + 0.5V$ ;  $V_{IH} = V_{DD} - 0.5V$ ; EA = RST = Port 0 = P1.6 = P1.7 = EW =  $V_{DD}$ ; STADC =  $V_{SS}$ ; CRX0 = 2.7V; CRX1 = 2.3V.
- The idle mode supply current is measured with all output pins disconnected; XTAL1 is driven with  $t_r = t_f = 10ns$ ;  $V_{IL} = V_{SS} + 0.5V$ ;  $V_{IH} = V_{DD} - 0.5V$ ; Port 0 = P1.6 = P1.7 = EW =  $V_{DD}$ ; EA = RST = STADC =  $V_{SS}$ ; CRX0 = 2.7V; CRX1 = 2.3V.
- The idle and sleep mode supply current is measured with all output pins disconnected; XTAL1 is driven with  $t_r = t_f = 10ns$ ;  $V_{IL} = V_{SS} + 0.5V$ ;  $V_{IH} = V_{DD} - 0.5V$ ; Port 0 = P1.6 = P1.7 = EW = CRX0 =  $V_{DD}$ ; EA = RST = STADC = CRX1 =  $V_{SS}$ ; CAN: register 6 = 00H, register 7 = 12H, register 8 = 02H, register 0 = 20H, wait 15 $t_{CY}$ , register 1 = 10H, wait for bit Sleep = 1.
- The power-down current is measured with all output pins disconnected; Port 0 = P1.6 = P1.7 = EW = CRX0 =  $V_{DD}$ ; EA = RST = STADC = CRX1 = XTAL1 =  $V_{SS}$ . Window devices have to be covered.
- Under steady state (non-transient) conditions,  $I_{OL}$  must be limited externally as follows:  
Maximum  $I_{OL}$  per 8 bit port  
Port 0: 36mA  
Port 1: 32mA  
Ports 2, 3 and 4: 15mA  
Maximum  $I_{OL}$  for all output pins: 71mA  
If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are guaranteed to sink current greater than the listed test conditions.
- Capacitive loads on Port 0 and Port 2 may degrade the LOW level output voltage of ALE, Port 1 and Port 3. During a 1-to-0 transition on the Port 0 and Port 2 pins and a capacitive load > 100pF, the ALE LOW level may exceed 0.8V. In the case that it is necessary to connect ALE to a Schmitt trigger input respectively, use an address latch with a Schmitt trigger STROBE input.
- Capacitive loads on Port 0 and Port 2 may cause a HIGH level voltage degradation of ALE and PSEN below 0.9 $V_{DD}$  during the address bits are stabilizing.
- $t_{CY} = 12t_{CLK}$  is the machine cycle time.
- $AV_{REF+} = 5.12V$ ;  $AV_{REF-} = 0V$ ;  $AV_{DD} = 5.0V$ .
- The differential non-linearity ( $DL_n$ ) is the difference between the actual step width and the ideal step width.
- The ADC is monotonic, there are no missing codes.
- The integral non-linearity ( $IL_n$ ) is the peak difference between the center of the steps of the actual and the ideal transfer curve after appropriate adjustment of gain and offset error.
- The offset error ( $OS_n$ ) is the absolute difference between the straight line which fits the actual transfer curve after removing gain error, and a straight line which fits the ideal transfer curve. The offset error is constant at every point of the actual transfer curve.
- The gain error ( $G_n$ ) is the relative difference in percent between the straight line fitting the actual transfer curve after removing offset error and the straight line which fits the ideal transfer curve. The gain error is constant at every point on the transfer curve.
- The absolute voltage error ( $A_n$ ) is the maximum difference between the center of the steps of the actual transfer curve of the not calibrated ADC and the ideal transfer curve.
- Not tested during production.
- Source current for the CTX0, CTX1 outputs together.

8-bit microcontroller with on-chip CAN

## P8xCE598

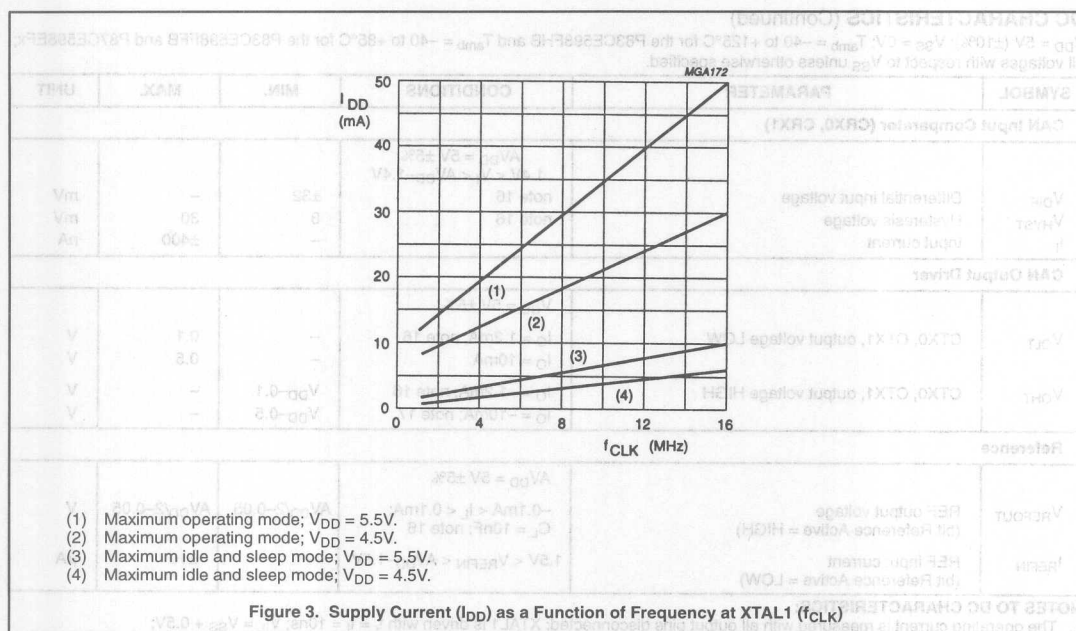
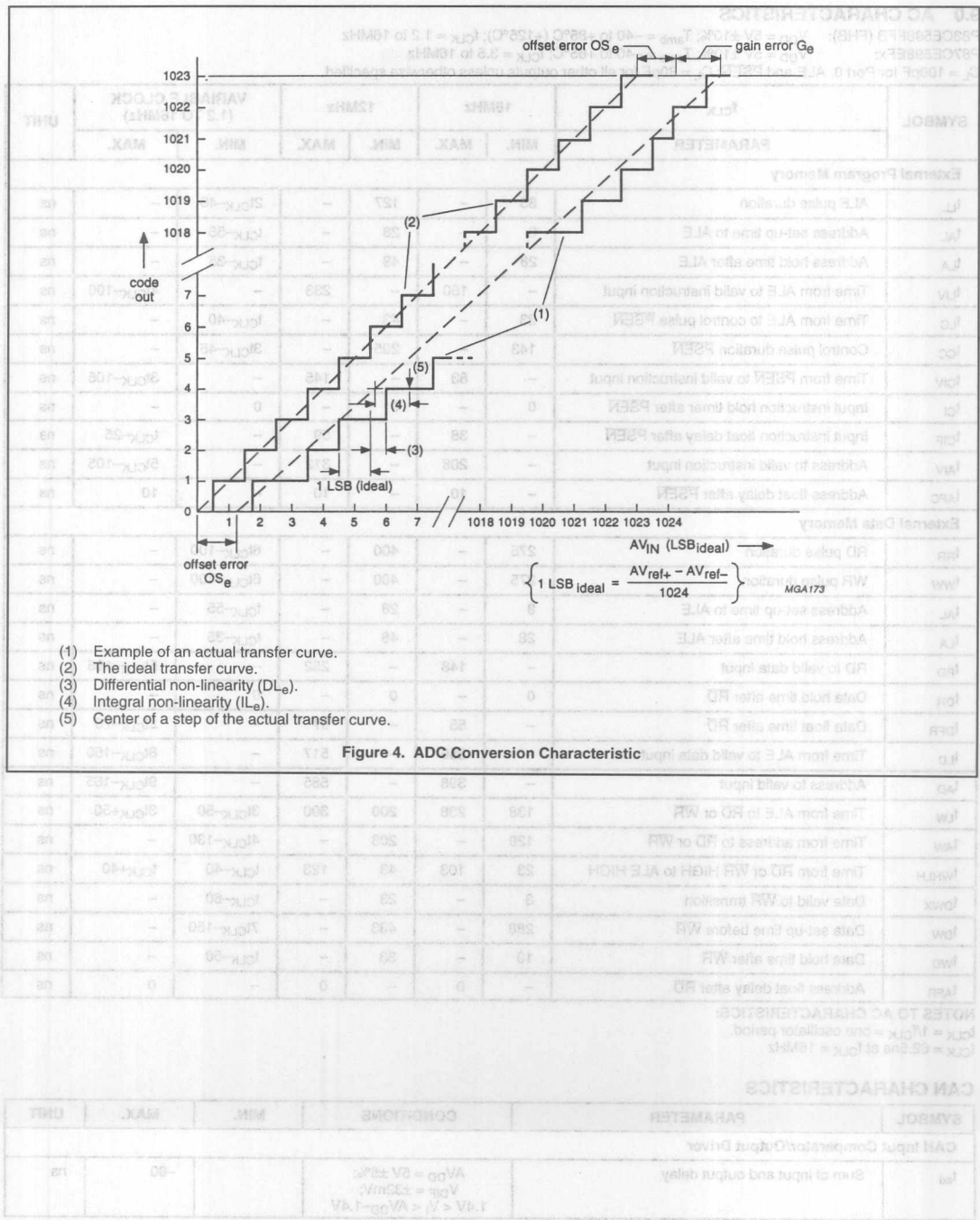


Figure 3. Supply Current ( $I_{DD}$ ) as a Function of Frequency at XTAL1 ( $f_{CLK}$ )

8-bit microcontroller with on-chip CAN P8xCE598





## 8-bit microcontroller with on-chip CAN

## P8xCE598

## 9.0 AC CHARACTERISTICS

P83CE598FFB (FHB):  $V_{DD} = 5V \pm 10\%$ ;  $T_{amb} = -40$  to  $+85^\circ C$  ( $+125^\circ C$ );  $f_{CLK} = 1.2$  to  $16MHz$ P87CE598EFx:  $V_{DD} = 5V \pm 10\%$ ;  $T_{amb} = -40$  to  $+85^\circ C$ ;  $f_{CLK} = 3.5$  to  $16MHz$  $C_L = 100pF$  for Port 0, ALE and PSEN;  $C_L = 80pF$  for all other outputs unless otherwise specified.

SYMBOL	$f_{CLK}$	16MHz		12MHz		VARIABLE CLOCK (1.2 TO 16MHz)		UNIT
	PARAMETER	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
External Program Memory								
$t_{LL}$	ALE pulse duration	85	—	127	—	$2t_{CLK}-40$	—	ns
$t_{AL}$	Address set-up time to ALE	8	—	28	—	$t_{CLK}-55$	—	ns
$t_{LA}$	Address hold time after ALE	28	—	48	—	$t_{CLK}-35$	—	ns
$t_{LIV}$	Time from ALE to valid instruction input	—	150	—	233	—	$4t_{CLK}-100$	ns
$t_{LC}$	Time from ALE to control pulse $\overline{PSEN}$	23	—	43	—	$t_{CLK}-40$	—	ns
$t_{CC}$	Control pulse duration $\overline{PSEN}$	143	—	205	—	$3t_{CLK}-45$	—	ns
$t_{CIV}$	Time from $\overline{PSEN}$ to valid instruction input	—	83	—	145	—	$3t_{CLK}-105$	ns
$t_{CI}$	Input instruction hold timer after $\overline{PSEN}$	0	—	0	—	0	—	ns
$t_{CIF}$	Input instruction float delay after $\overline{PSEN}$	—	38	—	59	—	$t_{CLK}-25$	ns
$t_{AIV}$	Address to valid instruction input	—	208	—	312	—	$5t_{CLK}-105$	ns
$t_{AFC}$	Address float delay after $\overline{PSEN}$	—	10	—	10	—	10	ns
External Data Memory								
$t_{RR}$	RD pulse duration	275	—	400	—	$6t_{CLK}-100$	—	ns
$t_{WW}$	WR pulse duration	275	—	400	—	$6t_{CLK}-100$	—	ns
$t_{AL}$	Address set-up time to ALE	8	—	28	—	$t_{CLK}-55$	—	ns
$t_{LA}$	Address hold time after ALE	28	—	48	—	$t_{CLK}-35$	—	ns
$t_{RD}$	RD to valid data input	—	148	—	252	—	$5t_{CLK}-165$	ns
$t_{DR}$	Data hold time after $\overline{RD}$	0	—	0	—	0	—	ns
$t_{DFR}$	Data float time after $\overline{RD}$	—	55	—	97	—	$2t_{CLK}-70$	ns
$t_{LD}$	Time from ALE to valid data input	—	350	—	517	—	$8t_{CLK}-150$	ns
$t_{AD}$	Address to valid input	—	398	—	585	—	$9t_{CLK}-165$	ns
$t_{LW}$	Time from ALE to $\overline{RD}$ or $\overline{WR}$	138	238	200	300	$3t_{CLK}-50$	$3t_{CLK}+50$	ns
$t_{AW}$	Time from address to $\overline{RD}$ or $\overline{WR}$	120	—	203	—	$4t_{CLK}-130$	—	ns
$t_{WHLH}$	Time from $\overline{RD}$ or $\overline{WR}$ HIGH to ALE HIGH	23	103	43	123	$t_{CLK}-40$	$t_{CLK}+40$	ns
$t_{DWX}$	Data valid to $\overline{WR}$ transition	3	—	23	—	$t_{CLK}-60$	—	ns
$t_{DW}$	Data set-up time before $\overline{WR}$	288	—	433	—	$7t_{CLK}-150$	—	ns
$t_{WD}$	Data hold time after $\overline{WR}$	13	—	33	—	$t_{CLK}-50$	—	ns
$t_{AFR}$	Address float delay after $\overline{RD}$	—	0	—	0	—	0	ns

## NOTES TO AC CHARACTERISTICS:

 $t_{CLK} = 1/f_{CLK}$  = one oscillator period. $t_{CLK} = 62.5ns$  at  $f_{CLK} = 16MHz$ 

## CAN CHARACTERISTICS

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
<b>CAN Input Comparator/Output Driver</b>					
$t_{sd}$	Sum of input and output delay	$AV_{DD} = 5V \pm 5\%$ ; $V_{DIF} = \pm 32mV$ ; $1.4V < V_I < AV_{DD}-1.4V$		-60	ns

## 8-bit microcontroller with on-chip CAN

## P8xCE598

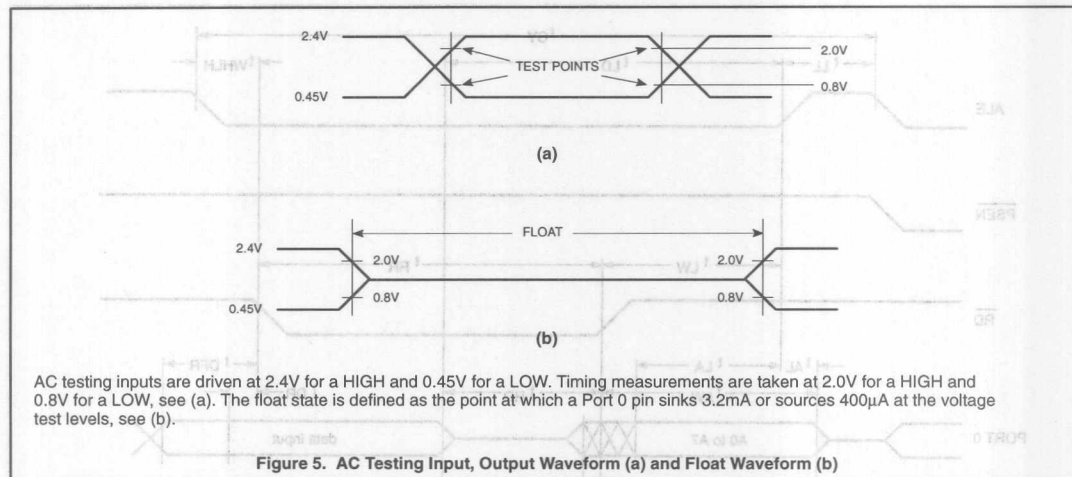


Figure 5. AC Testing Input, Output Waveform (a) and Float Waveform (b)

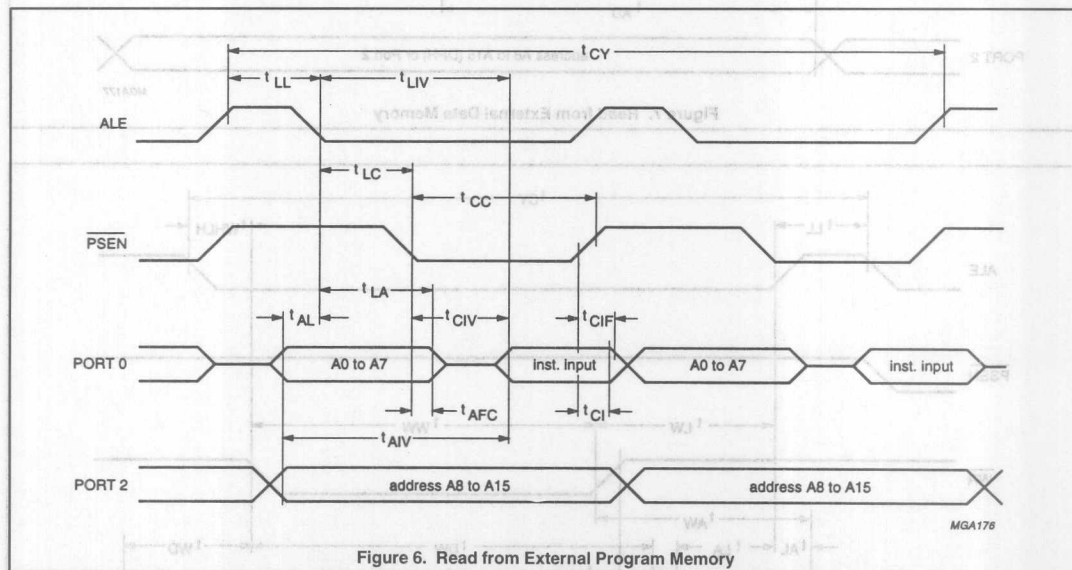
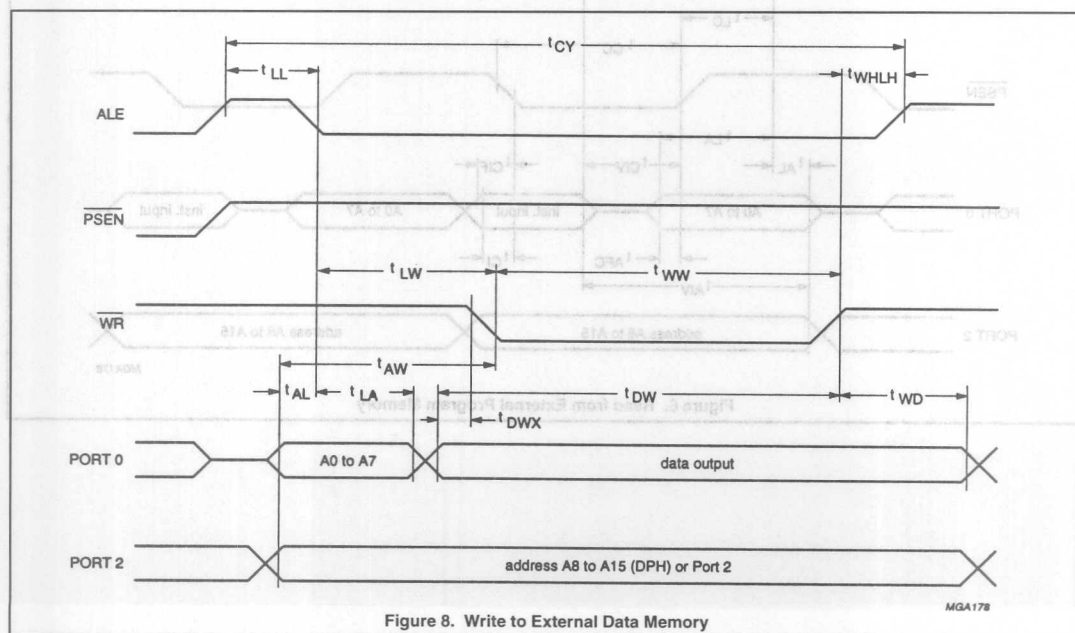
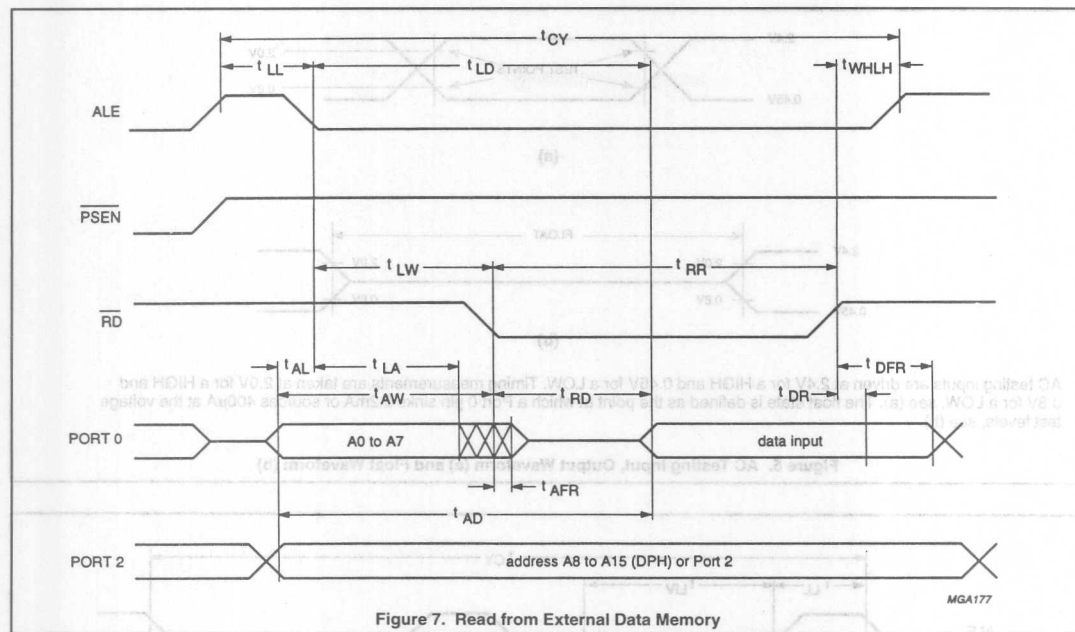


Figure 6. Read from External Program Memory

## 8-bit microcontroller with on-chip CAN

## P8xCE598



8-bit microcontroller with on-chip CAN P8xCE598

Table 1. External Clock Drive XTAL1

SYMBOL	PARAMETER	VARIABLE CLOCK (f <sub>CLK</sub> = 1.2/3.5 to 16MHz)		UNIT
		MIN.	MAX.	
t <sub>CLK</sub>	Oscillator clock period P83CE598	62.5	833.3	ns
t <sub>CLK</sub>	Oscillator clock period P87CE598	62.5	285.7	ns
t <sub>HIGH</sub>	HIGH time	20	t <sub>CLK</sub> - t <sub>LOW</sub>	ns
t <sub>LOW</sub>	LOW time	20	t <sub>CLK</sub> - t <sub>HIGH</sub>	ns
t <sub>r</sub>	Rise time	—	20	ns
t <sub>f</sub>	Fall time	—	20	ns

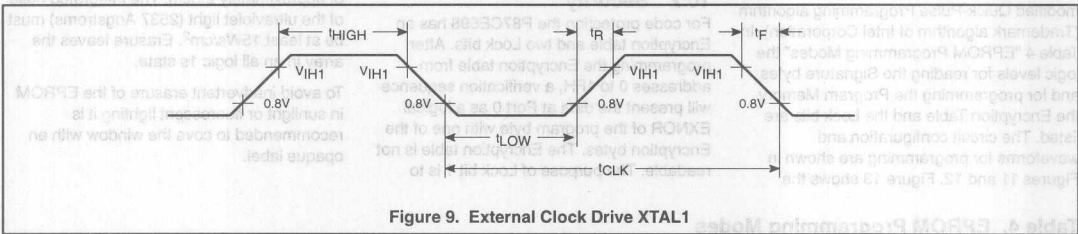
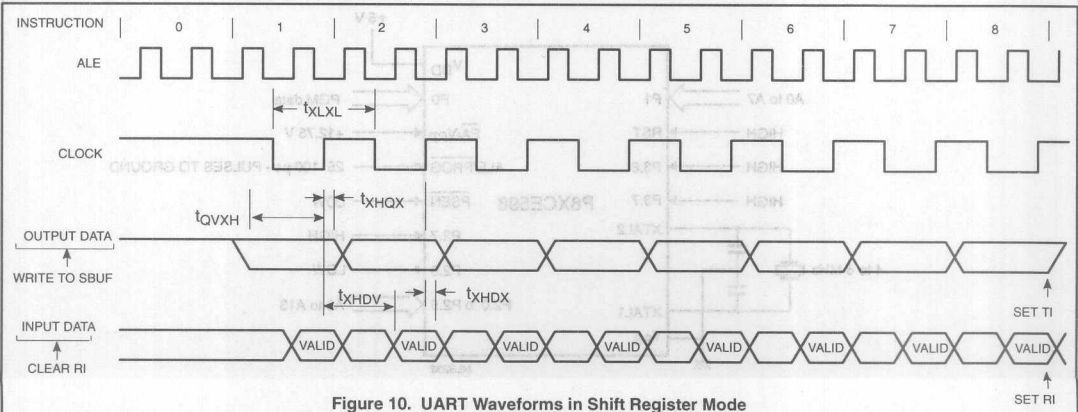


Table 2. UART Timing in Shift Register Mode (see Figure 10)

SYMBOL	PARAMETER	f <sub>CLK</sub>		16MHz		12MHz		VARIABLE CLOCK		UNIT
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
t <sub>XLXL</sub>	Serial Port clock cycle timing	0.75	—	1.0	—	12t <sub>CLK</sub>	—	—	—	ms
t <sub>QVXH</sub>	Output data setup to clock rising edge	492	—	700	—	10t <sub>CLK</sub> -133	—	—	—	ns
t <sub>XHQX</sub>	Output data hold after clock rising edge	8.0	—	50	—	2t <sub>CLK</sub> -117	—	—	—	ns
t <sub>XHDX</sub>	Input data hold after clock rising edge	0	—	0	—	0	—	—	—	ns
t <sub>XHDV</sub>	Clock rising edge to input data valid	—	492	—	700	—	—	10t <sub>CLK</sub> -133	—	ns



# 8-bit microcontroller with on-chip CAN P8xCE598

## 10.0 EPROM

### CHARACTERISTICS

The P8xCE598 has an on-chip 32 kbyte EPROM for fast and flexible controller software development. Two versions are available; one in a ceramic QFP package with window for UV-erasure (P87CE598 EFQ) and a low-cost OTP (One Time Programming) version in a Plastic Quad Flat Pack (PQFP) package (87CE598EFB), which is not erasable.

### 10.1 Programming and Verification

The 87CE598 is programmed by using a modified Quick-Pulse Programming algorithm (Trademark algorithm of Intel Corporation). In Table 4 "EPROM Programming Modes" the logic levels for reading the Signature bytes and for programming the Program Memory the Encryption Table and the Lock bits are listed. The circuit configuration and waveforms for programming are shown in Figures 11 and 12. Figure 13 shows the

circuit configuration for code data verification.

Note that programming and verification is done with an oscillator frequency of 4 to 6 MHz. The two Signature bytes identifying the device as a P87CE598 manufactured by Philips are located as shown in Table 3.

Table 3. Signature Bytes

ADDRESS	CONTENT	MEANING
30H	15H	Philips
31H	9CH	P87CE598

### 10.2 Security

For code protection the P87CE598 has an Encryption table and two Lock bits. After programming the Encryption table from addresses 0 to 1FH, a verification sequence will present the data at Port 0 as a logical EXNOR of the program byte with one of the Encryption bytes. The Encryption table is not readable. The purpose of Lock bit 1 is to

inhibit programming. It is impossible to fetch code bytes from the internal Program Memory with MOV<sub>C</sub>-instructions executed from external memory. However, verification of the EPROM and programming of the second Lock bit is still possible. Lock bit 2 additionally disables verification of the program code. Note that for further security the EA input level of the P8xCE598 is latched during Reset.

### 10.3 Erasure

A controlled erasure is done by exposing the EPROM window to an ultraviolet lamp of  $12\mu\text{W}/\text{cm}^2$  for 20 to 39 minutes at a distance of approximately 2.5cm. The integrated dose of the ultraviolet light (2537 Angstroms) must be at least  $15\text{Ws}/\text{cm}^2$ . Erasure leaves the array in an all logic 1s state.

To avoid inadvertent erasure of the EPROM in sunlight or fluorescent lighting it is recommended to cover the window with an opaque label.

Table 4. EPROM Programming Modes

MODE	RST	PSEN	ALE/PROG	EA/V <sub>PP</sub>	P2.7	P2.6	P3.7	P3.6
Read Signature	HIGH	LOW	HIGH	HIGH	LOW	LOW	LOW	LOW
program code data	HIGH	LOW	LOW (note 1)	V <sub>PP</sub> (note 1)	HIGH	LOW	HIGH	HIGH
Verify code data	HIGH	LOW	HIGH	HIGH	LOW	LOW	HIGH	HIGH
Pgm.Encrypt.table	HIGH	LOW	LOW (note 1)	V <sub>PP</sub> (note 1)	HIGH	LOW	HIGH	LOW
Program Lock bit 1	HIGH	LOW	LOW (note 1)	V <sub>PP</sub> (note 1)	HIGH	HIGH	HIGH	HIGH
Program Lock bit 2	HIGH	LOW	LOW (note 1)	V <sub>PP</sub> (note 1)	HIGH	HIGH	LOW	LOW

#### NOTE:

1. ALE/PROG receives 25 programming pulses while V<sub>PP</sub> is held at  $12.75\text{V} \pm 0.25\text{V}$ . Each programming pulse is LOW for  $100\mu\text{s} \pm 10\mu\text{s}$  and HIGH for at least  $10\mu\text{s}$ .

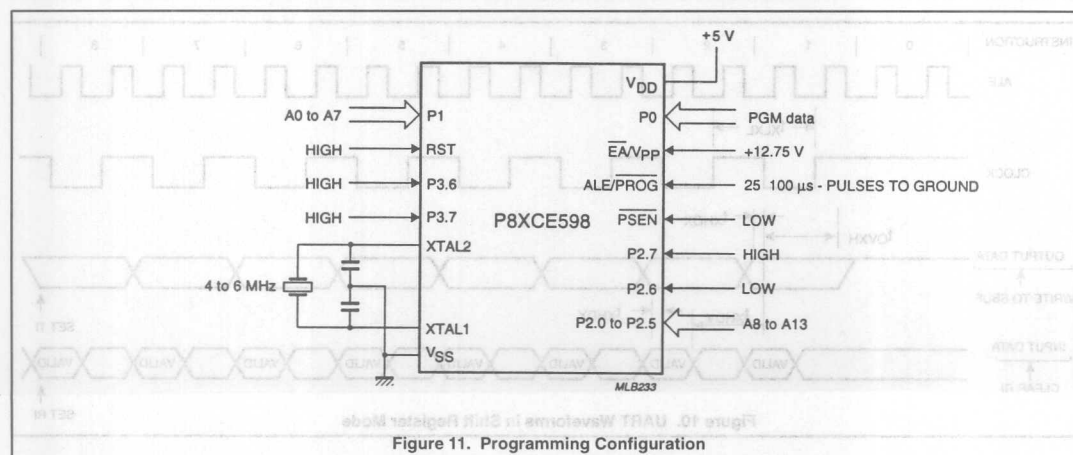
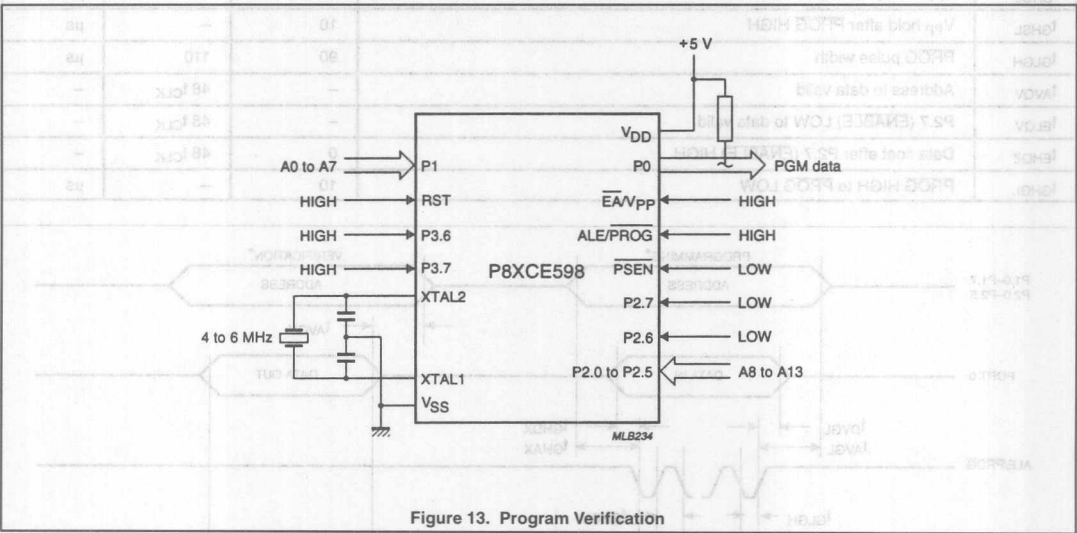
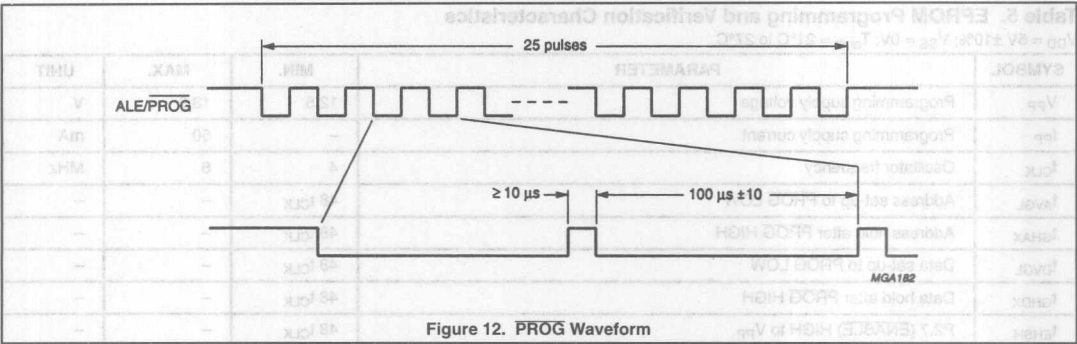


Figure 11. Programming Configuration



8-bit microcontroller with on-chip CAN

P8xCE598

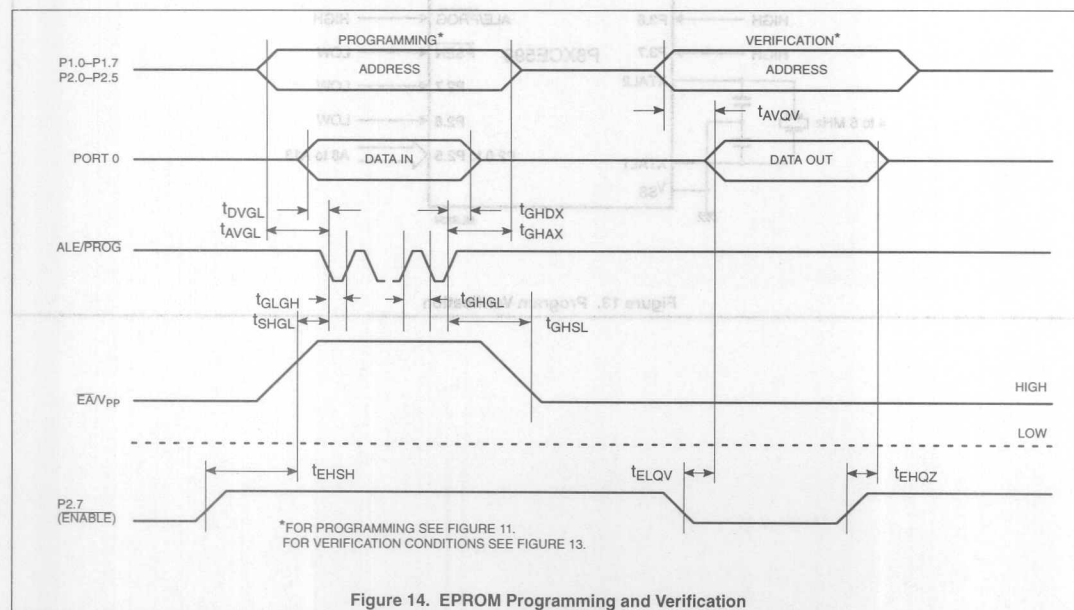


## 8-bit microcontroller with on-chip CAN

## P8xCE598

**Table 5. EPROM Programming and Verification Characteristics** $V_{DD} = 5V \pm 10\%$ ;  $V_{SS} = 0V$ ;  $T_{amb} = 21^{\circ}C$  to  $27^{\circ}C$ 

SYMBOL	PARAMETER	MIN.	MAX.	UNIT
$V_{PP}$	Programming supply voltage	12.5	13.0	V
$I_{PP}$	Programming supply current	—	50	mA
$f_{CLK}$	Oscillator frequency	4	6	MHz
$t_{AVGL}$	Address set-up to $\overline{PROG}$ LOW	48 $t_{CLK}$	—	—
$t_{GHAX}$	Address hold after $\overline{PROG}$ HIGH	48 $t_{CLK}$	—	—
$t_{DVGL}$	Data set-up to $\overline{PROG}$ LOW	48 $t_{CLK}$	—	—
$t_{GHDX}$	Data hold after $\overline{PROG}$ HIGH	48 $t_{CLK}$	—	—
$t_{EHS}$	P2.7 (ENABLE) HIGH to $V_{PP}$	48 $t_{CLK}$	—	—
$t_{SHGL}$	$V_{PP}$ set-up to $\overline{PROG}$ LOW	10	—	$\mu s$
$t_{GHSL}$	$V_{PP}$ hold after $\overline{PROG}$ HIGH	10	—	$\mu s$
$t_{GLGH}$	$\overline{PROG}$ pulse width	90	110	$\mu s$
$t_{AVQV}$	Address to data valid	—	48 $t_{CLK}$	—
$t_{ELQV}$	P2.7 (ENABLE) LOW to data valid	—	48 $t_{CLK}$	—
$t_{EHQZ}$	Data float after P2.7 (ENABLE) HIGH	0	48 $t_{CLK}$	—
$t_{GHGL}$	$\overline{PROG}$ HIGH to $\overline{PROG}$ LOW	10	—	$\mu s$



8-bit microcontroller with on-chip CAN

P8xCE598

11.0 CAN APPLICATION INFORMATION

11.1 Latency Time Requirements

Real-time applications require the ability to process and transfer information in a limited and predetermined period of time. If knowing this total time and the time required to process the information, the (maximum allowed) transfer delay time is given. It is measured from the initiation of the transfer up to the signalling of reception. For instance, this is the period of time between programming the CAN Command Register bit 0 (Transmission Request) to HIGH and the time getting an interrupt at a receiving CAN-device (due to the reception of the respective message).

The maximum allowed bit-time ( $t_{BIT}$ ) due to latency time requirements can be calculated as:

(1)  $t_{BIT} \leq t_{MAX\ TRANSFER\ TIME} / (n_{BIT,\ MAX\ LATENCY} + n_{BIT,\ MESSAGE})$

- $t_{MAX\ TRANSFER\ TIME}$ :  
 $n_{BIT,\ MAX\ LATENCY}$ :  
 $n_{BIT,\ MESSAGE}$ :
- the maximum allowed transfer delay time (application-specific);

the maximum latency time (in terms of number of bits), which depends on the actual state of the CAN network (e.g., another message already on the network);

the number of bits of a message; it varies with the number of transferred data bytes  $n_{DATA\ BYTES}$  (0..8) and Stuffbits like:

(2)  $44 + 8 \cdot n_{DATA\ BYTES} \leq n_{BIT,\ MESSAGE} \leq n_{BIT,\ MESSAGE} \leq 52 + 10 \cdot n_{DATA\ BYTES}$

Example:

For the calculation of  $n_{BIT,\ MAX\ LATENCY}$  the following is assumed (the term "our message" refers to that one the latency time is calculated for):

- Since at maximum one-bit-time ago another CAN Controller is transmitting
- A single error occurs during the transmission of that message preceding ours, leading to the additional transfer of one Error Frame.
- "Our message" has the highest priority.

giving:

(3)  $n_{BIT,\ MAX\ LATENCY} \geq 44 + 8 \cdot n_{DATA\ BYTES,\ WORST\ CASE} + 18;$   
 $n_{BIT,\ MAX\ LATENCY} \leq 52 + 10 \cdot n_{DATA\ BYTES,\ WORST\ CASE} + 18;$

The additional 18 bits are due to the Error Frame and the Intermission Field preceding our message.  $n_{DATA\ BYTES,\ WORST\ CASE}$  denotes the number of data bytes contained by the longest message being used in a given CAN network.

Here an example is provided to help calculating the maximum bit-time:

EXAMPLE: MAXIMUM BIT-TIME CALCULATION

STATEMENT	COMMENTS
$t_{MAX\ TRANSFER\ TIME} = 10ms$	assumption;
$n_{DATA\ BYTES,\ WORST\ CASE} = 6$	longest message in that network; assumption;
$n_{DATA\ BYTES} = 4$	our message; assumption;
$n_{BIT,\ MAX\ LATENCY} \leq 130$	using equation (3);
$n_{MESSAGE} \leq 92$	using equation (2);
$t_{BIT} \leq 10ms / (130 + 92) = 45\mu s$	using equation (1).

## 8-bit microcontroller with on-chip CAN

## P8xCE598

## 11.2 Connecting a P8xCE598 to a Bus Line (Physical Layer)

## On-Chip Transceiver

The P8xCE598 features an on-chip differential transceiver including output driver and input comparator both being configurable (see Figure 15). Therefore it supports many types of common transmission media such as:

- Single wire bus line
- Two-wire bus line (differential)
- Optical cable bus line.

The P8xCE598 can directly drive a differential bus line. An example is given in Figure 16 for a bus line having a characteristic impedance of 120Ω. Direct interfacing to the bus line is well suited for applications with limited requirements concerning electromagnetic susceptibility, wiring failure tolerance and protection against transients.

## Transceiver for In-Vehicle Communication

Figure 17 shows a versatile transceiver implementation designed for automotive applications. It features a bit rate of up to 1 Mbit/s and dissipates low power during standby (1.4mA), thus it is suitable also for

applications requiring a sleep mode function with system activation via the bus line. The transceiver provides an extended common mode range for high electromagnetic susceptibility performance.

Two external driver transistors amplify the output current to 35mA typically and provide protection against overvoltage conditions on the bus line (e.g., due to an accidental short-circuit between a bus wire and battery voltage). The serial diodes prevent, in combination with the transistors, the bus from being blocked in case of a bus not powered. More than 32 nodes may be connected to the bus line.

## Detection and Handling of Bus Wiring Failures

Using the P8xCE598 a superior wiring failure tolerance and detection performance can be achieved. This requires both bus lines to be mutually decoupled as shown in Figure 18. Each bus wire is biased separately to a reference voltage of  $AV_{DD}/2$ .

The diodes suppress reverse current in case of a termination circuit being not properly powered or a bus line being short-circuited, i.e., to a voltage higher than 5V. Applying this bus termination circuit the following wiring

failures on the bus are detectable and can be handled:

- Interruption of one bus wire at any location
- Short-circuit of one bus wire to ground or to battery voltage
- Short-circuit between the bus wires.

A bus failure can be detected, e.g., by a drop out of a status message, regularly being transmitted on the bus. If a bus wire is corrupted, the following actions have to be taken:

- Switch the corresponding comparator input over to a reference voltage of  $AV_{DD}/2$
- Disable the corresponding output driver stage.

As a consequence, communication will continue on that bus wire not being corrupted. The required reference voltage and the switches for the comparator inputs are provided on-chip. An output driver stage can be disabled by reconfiguration of the on-chip output driver (reprogramming of the Output Control register of the P8xCE598). To find out which of the bus wires is corrupted, a heuristic method is applied.

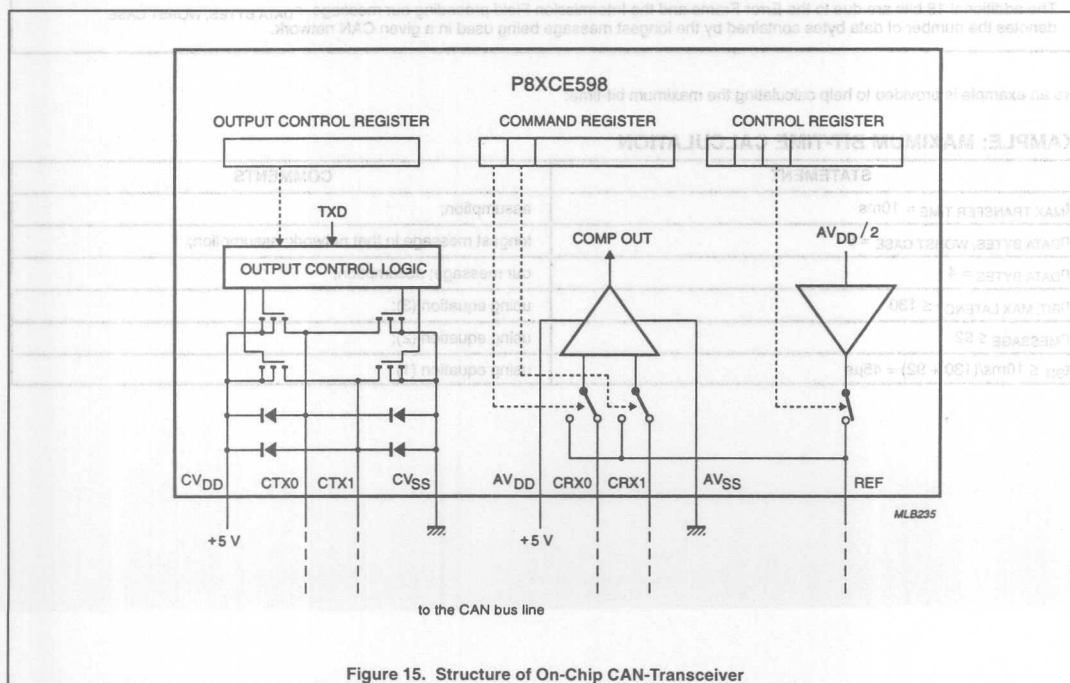
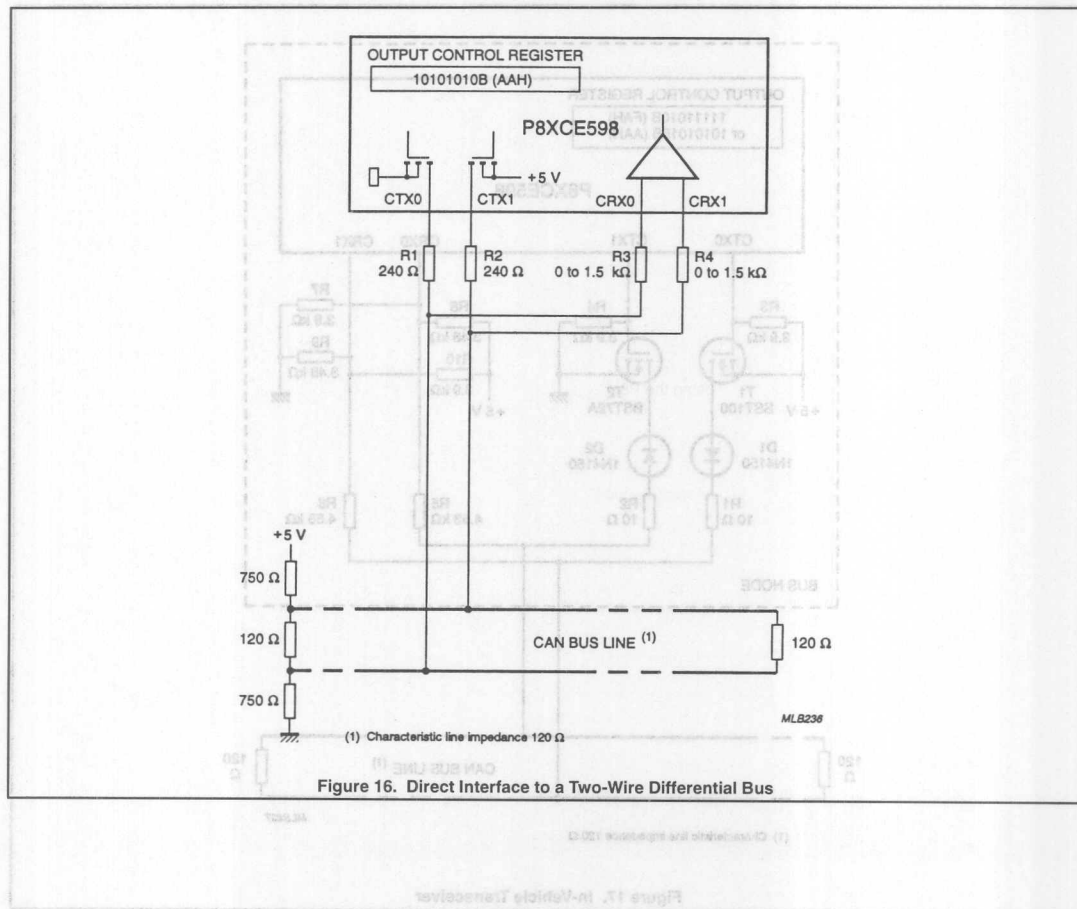


Figure 15. Structure of On-Chip CAN-Transceiver

# 8-bit microcontroller with on-chip CAN

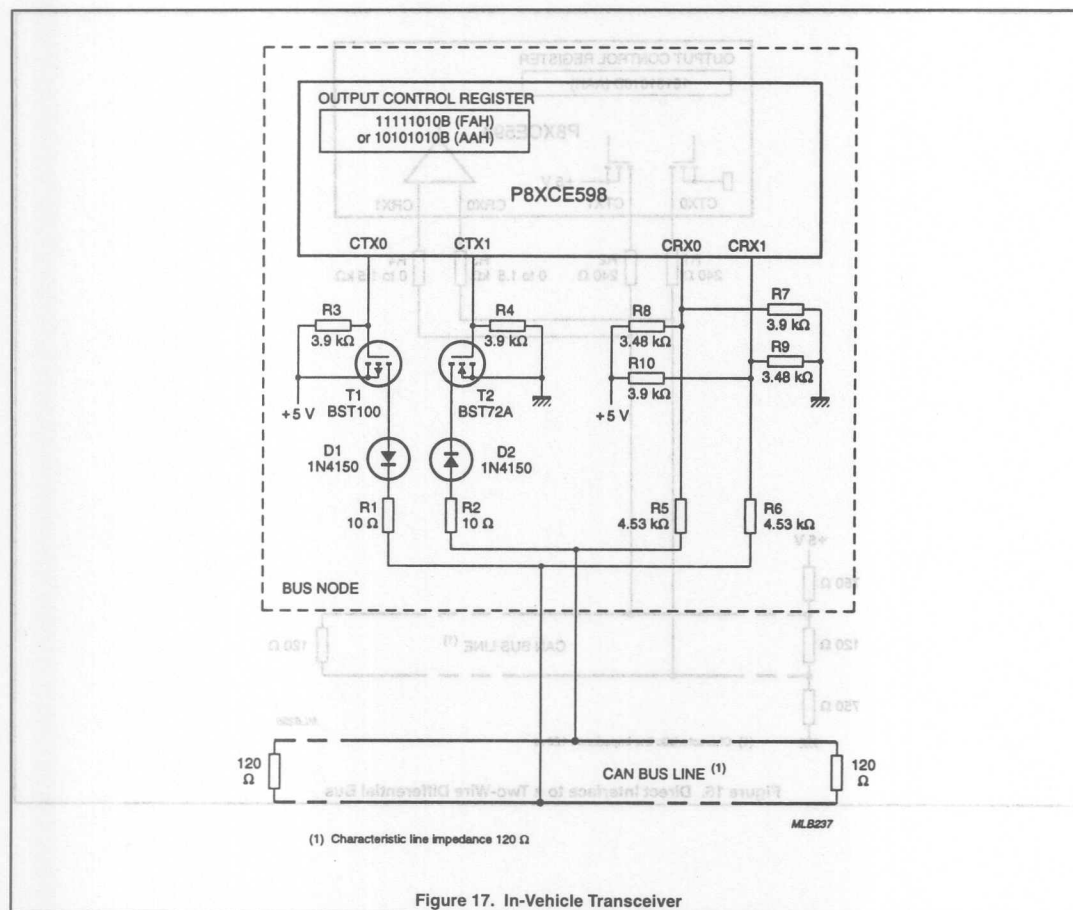
## P8xCE598





## 8-bit microcontroller with on-chip CAN

## P8xCE598



## 8-bit microcontroller with on-chip CAN

## P8xCE598

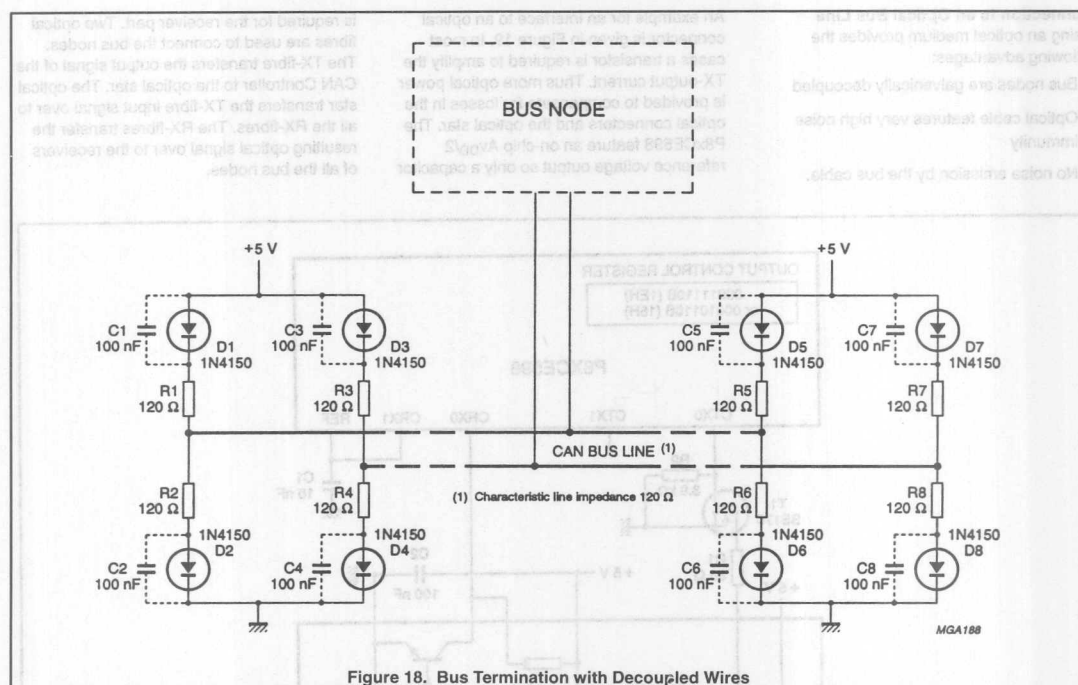


Figure 18. Bus Termination with Decoupled Wires

## 8-bit microcontroller with on-chip CAN

## P8xCE598

**Connection to an Optical Bus Line**

Using an optical medium provides the following advantages:

- Bus nodes are galvanically decoupled
- Optical cable features very high noise immunity
- No noise emission by the bus cable.

An example for an interface to an optical connector is given in Figure 19. In most cases a transistor is required to amplify the TX-output current. Thus more optical power is provided to compensate for losses in the optical connectors and the optical star. The P8xCE598 feature an on-chip  $A_{VDD/2}$  reference voltage output so only a capacitor

is required for the receiver part. Two optical fibres are used to connect the bus nodes. The TX-fibre transfers the output signal of the CAN Controller to the optical star. The optical star transfers the TX-fibre input signal over to all the RX-fibres. The RX-fibres transfer the resulting optical signal over to the receivers of all the bus nodes.

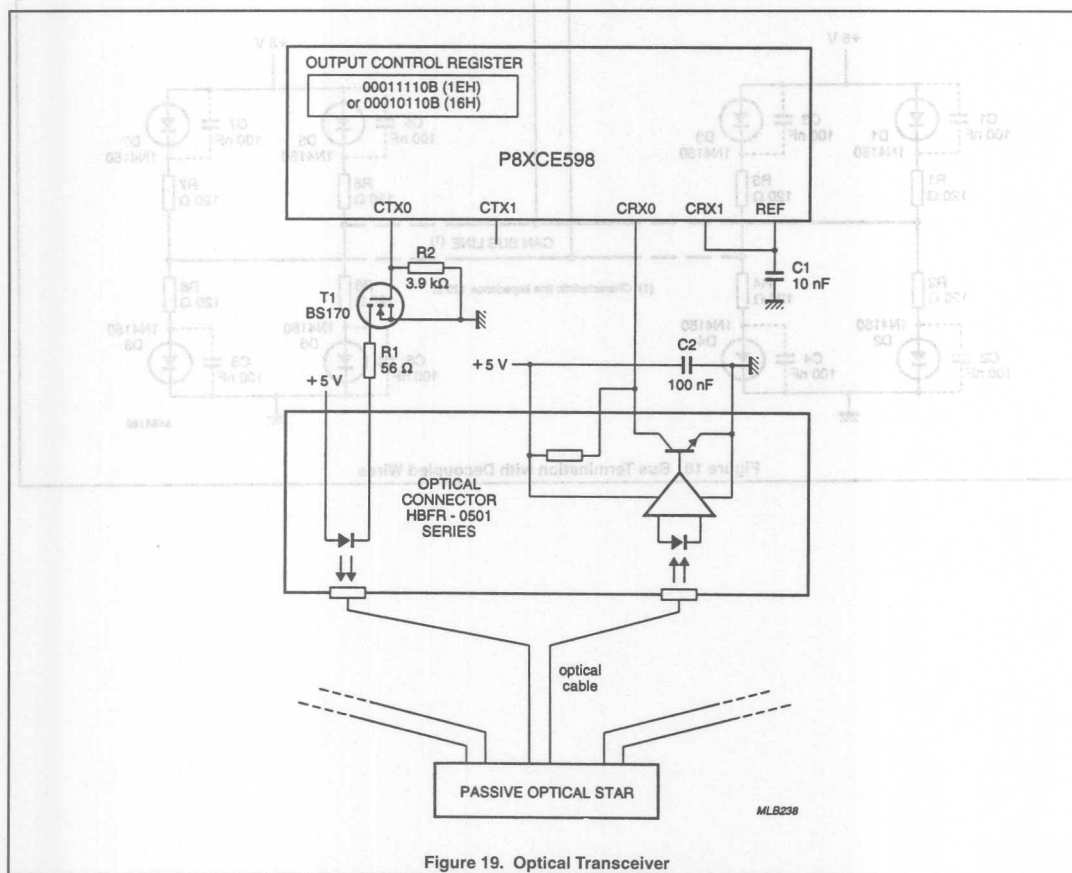


Figure 19. Optical Transceiver

## 8-bit microcontroller with on-chip CAN P8xCE598

## CAN Interrupt Handler Software Example

Software example for P8xCE598 CAN interrupt handler (including fast DMA transfer)

MC5-51 MACRO ASSEMBLER P8xCE598 CAN interrupt-handler.

```

LOC  OBJ  LINE SOURCE
-----
1  $TITLE (8XCE598 CAN interrupt-handler)
2  $NOSYMBOLS NOPAGING
3
4  ;*****
5  ;
6  ;Very fast receive-routine for the 8XCE598. It:
7  ;o      is embedded in the interrupt-handler for the CAN Controller,
8  ;o      uses the DMA-logic and
9  ;o      handles up to eight different messages
10 ;      (if these have the same leading 8 identifier-bits).
11 ;
12 ;To allow for faster receive-routine, it is assumed that all other routines
13 ;accessing the CAN Controller disable the interrupt of the CAN Controller
14 ;(IEN0.5) during their execution.
15 ;
16 ;Version: 1.0
17 ;Date: 12-April-90
18 ;Author: Bernhard Reckels
19 ;at: Philips Components Application Lab., Hamburg (PCALH)
20 ;*****
21 ;*****
22 ;*****
23 ;*****
24 ;initial stuff
25 ;*****
26 ;*****
27 ;equatas
28
29 ;addresses of Special Function Registers
30 CANADR EQU 0DBH
31 CANDAT EQU 0DAH
32 CANCON EQU 0D9H
33 CANSTA EQU 0D8H
34
35 ;commands for the CAN Controller / DMA logic
36 CAN_REF_REL EQU 00000100B ;Release Receive Buffer
37 CAN_RX_DMA EQU 80H + 22 ;Rx DMA-transfer
38
39 ;addresses of CAN Controller internal registers
40 CAN_REF EQU 20 ;1st address of Rx-buffer
41
42 ;masks
43 INT_FLAG_MASK EQU 00011111B ;all CAN's interrupt-flags
44 ID2_0_MASK EQU 11100000B ;only ID.2 ... ID.0 bits
45
46 ;jump-address for a CAN Controller interrupt
47 ;JUMP to 2BH
48 CSEG at 2BH
49 LJMP CAN_INT_HANDLER ;CAN's interrupt-vector
50
51 ;data storage
52
53 DSEG at 20H
54 CAN_INT_IMAGE: DS 1
55

```

## 8-bit microcontroller with on-chip CAN

## P8xCE598

```

-----
56 BSEG at 00H
0000 57 CAN_INT_RX: DBIT 1 ;=CAN_INT_IMAGE.0
0001 58 CAN_INT_TX: DBIT 1 ;=CAN_INT_IMAGE.1
0002 59 CAN_INT_KR: DBIT 1 ;=CAN_INT_IMAGE.2
0003 60 CAN_INT_OV: DBIT 1 ;=CAN_INT_IMAGE.3
0004 61 CAN_INT_WK: DBIT 1 ;=CAN_INT_IMAGE.4
62
63 ;*****
64 ;CAN Controller interrupt-handler
65 ;
66 ;Only the receive-interrupt is coded.
67 ;
68 ;*****
69
-----
70 CSEG at 080H
71
72 CAN_INT_HANDLER:
73
74 ;first save used registers
0080 C0D0 75 PUSH PSW
0082 C0E0 76 PUSH ACC
77
78 ;store the CAN Controller's Interrupt Register contents
79 ;(here: at a bit-addressable location).
80 ;This is necessary because after reading the Interrupt Register
81 ;its contents is cleared, but - on the other hand - several flags
82 ;may be set in coincidence.
0084 E5D9 83 MOV A, CANON
0086 541F 84 ANL A, #INT_FLAG_MASK ;only interrupt-flags
0088 F520 85 MOV CAN_INT_IMAGE, A
86
87
88 ;dispatcher-----
89 INT_TEST0:
008A 10000 90 JBC CAN_INT_RC, CAN_RX_SERV ;receive-interrupt ?
91
92 INT_TEST1:
93 ;here the dispatcher has to be completed according
94 ;to the application-specific requirements
95 ;...
96 ;...
97 ;end of dispatcher-----
98
99 ;Rx-serve-----
100 ;copy message (Data-Field only) from CAN- to CPU memory
101
102 CAN_RX_SERVE
103 ;read 2nd Descriptor-Byte from the Rx-Buffer (address 21)
008D 75DB15 104 MOV CANADR, #CAN_REF + 1
0090 E5DA 105 MOV A, CANDAT
106
107 ;determine the destination address in data-memory for the
108 ;message's Data-Field
0092 54E0 109 ANL A, #ID2_0_MASK ;use ID.2 ... ID.0 only
0094 C4 110 SWAP A
0095 03 111 RR A ;A = 4*ID.2 + 2*ID.1 + ID.0
112 ;this value is used as an index for an array of 8 bytes
113 ;containing the destination-addresses for the 8 different
114 ;messages. Note, that the #RX_ARRAY_START - RX_ARRAY_OFFSET
115 ;program counter-relative access to the array.
0096 2415 116 ADD A, #RX_ARRAY_START - RX_ARRAY_OFFSET
0098 83 117 MOVC A, @A + PC
118 RX_ARRAY_OFFSET:
119

```



## 8-bit microcontroller with on-chip CAN P8xCE598

```

120 ;if a message passes the acceptance-filter of the CAN
121 ;Controller, but the CPU doesn't need it, the array
122 ;entry's value may be set to zero indicating this.
123 ;The following <jz> instruction cares for this.
0099 6007 124 JZ      CAN_RX_READY
125
126 ;now copy the Data-Field (only) from CAN- to CPU memory
127 ;with the aid of the DMA-logic. note, that a TX-DMA is
128 ;performed when writing 8AH (DMA + address 10) into CANADR
129 ;and a RX-DMA is performed when writing 94H (DMA + address 20)
130 ;... 9DH (DMA + address 29) into CANADR. Here address 22 is
131 ;used to copy just the Data-Field.
009B F5D8 132 MOV     CANSTA,A      ;data-memory address
009D 75DB96 133 MOV     CANADR, #CAN_RX_DMA ;starts RX-DMA at address 22
134
135 ;the DMA-transfer is done in at maximum 2 instruction cycles.
136 ;During the transfer, neither the data-memory (RAM) nor one
137 ;of the SFRs CANADR, CANDAT, CANCON and
138 ;CANSTA may be accessed by the CPU.
139 ;For simplicity, two NOPs are used here.
00A0 00 140 NOP
00A1 00 141 NOP
142
143 ;after reading the Rx-Buffer it must be released back to
144 ;the CAN Controller. In coincidence, the Clear Overrun bit
145 ;(CANCON.3) may be set, regardless of an existing or
146 ;non-existing data overrun.
147 CAN_RX_READY:
00A2 75D904 148 MOV     CANCON, #CAN_RBF_REL
149
150 ;if no other interrupt-flag is set, the interrupt-handler
151 ;for the CAN Controller can be left. Otherwise further
152 ;services are required.
00A5 E520 153 MOV     A, CAN_INT_IMAGE
00A7 70E4 154 JNZ     INT_TEST1
155
156 ;no other service is required, so the interrupt-handler
157 ;is left.
00A9 D0E0 158 POP     ACC
00AB D0D0 159 POP     PSW
00AD 32 160 RETI
161 ;end of Rx-serve-----
162
163 ;here the array follows containing 8 destination-addresses
164 ;for up to 8 different messages to be received. The values
165 ;are fully application-specific (the values below show an
166 ;example only).
167 RX_ARRAY_START:
00AE E0 168 DB      0E0H      ;Rx-message #0
00AF 00 169 DB      000H      ;this message is not used
170 ;...
00B0 FA 171 DB      0FAH      ;RX-message #7, containing 6 data bytes
172
173 END

REGISTER BANK(S) USED: 0
ASSEMBLY COMPLETE, NO ERRORS FOUND

```

# CMOS single-chip 8-bit microcontroller 80C652/83C652

## DESCRIPTION

The P80C652/83C652 Single-Chip 8-Bit Microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 80C652/83C652 has the same instruction set as the 80C51. Three versions of the derivative exist:

83C652 — 8k bytes mask programmable ROM

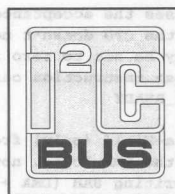
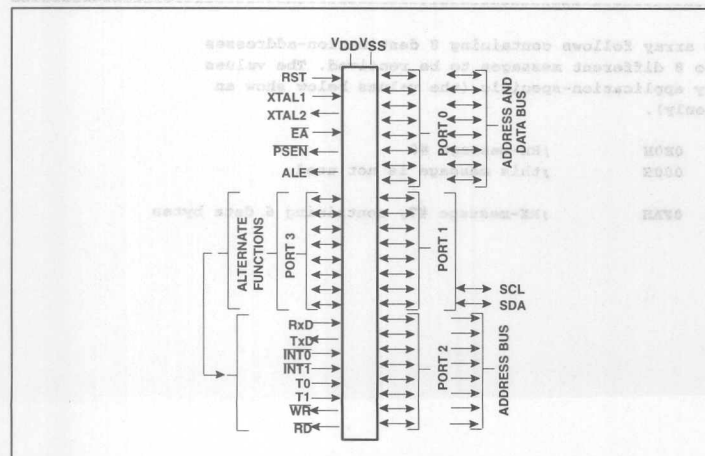
80C652 — ROMless version

80C652 — EPROM version (described in a separate chapter)

This device provides architectural enhancements that make it applicable in a variety of applications for general control systems. The 8XC652 contains a non-volatile 8k × 8 read-only program memory, a volatile 256 × 8 read/write data memory, four 8-bit I/O ports, two 16-bit timer/event counters (identical to the timers of the 80C51), a multi-source, two-priority-level, nested interrupt structure, an I<sup>2</sup>C interface, UART and on-chip oscillator and timing circuits. For systems that require extra capability, the 8XC652 can be expanded using standard TTL compatible memories and logic.

The device also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of over 100 instructions: 49 one-byte, 45 two-byte and 17 three-byte. With a 16(24)MHz crystal, 58% of the instructions are executed in 0.75(0.5)μs and 40% in 1.5(1)μs. Multiply and divide instructions require 3(2)μs.

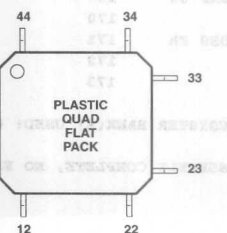
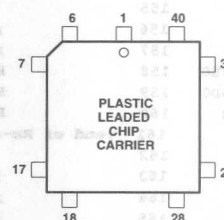
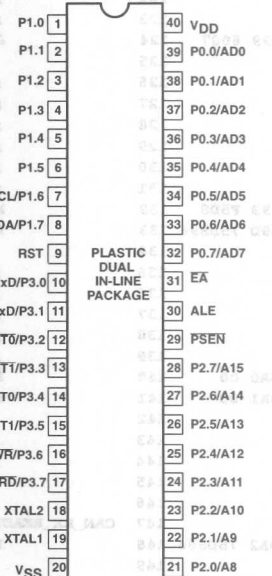
## LOGIC SYMBOL



## FEATURES

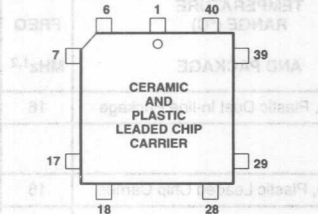
- 80C51 central processing unit
- 8k × 8 ROM expandable externally to 64k bytes
- 256 × 8 RAM, expandable externally to 64k bytes
- Two standard 16-bit timer/counters
- Four 8-bit I/O ports
- I<sup>2</sup>C-bus serial I/O port with byte oriented master and slave functions
- Full-duplex UART facilities
- Power control modes
  - Idle mode
  - Power-down mode
- ROM code protection
- Extended frequency range: 1.2 to 24 MHz
- Three operating ambient temperature ranges:
  - 0 to +70°C
  - -40 to +85°C
  - -40 to +125°C

## PIN CONFIGURATIONS



CMOS single-chip 8-bit microcontroller 80C652/83C652

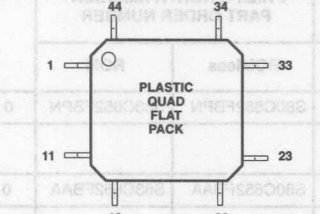
CERAMIC AND PLASTIC LEADED CHIP CARRIER PIN FUNCTIONS



CERAMIC AND PLASTIC LEADED CHIP CARRIER

Pin	Function	Pin	Function
1	NC	23	NC
2	P1.0	24	P2.0/A8
3	P1.1	25	P2.1/A9
4	P1.2	26	P2.2/A10
5	P1.3	27	P2.3/A11
6	P1.4	28	P2.4/A12
7	P1.5	29	P2.5/A13
8	P1.6/SCL	30	P2.6/A14
9	P1.7/SDA	31	P2.7/A15
10	RST	32	PSEN
11	P3.0/RxD	33	ALE
12	NC	34	NC
13	P3.1/TxD	35	EA
14	P3.2/INT0	36	P0.7/AD7
15	P3.3/INT1	37	P0.6/AD6
16	P3.4/T0	38	P0.5/AD5
17	P3.5/T1	39	P0.4/AD4
18	P3.6/WR	40	P0.3/AD3
19	P3.7/RD	41	P0.2/AD2
20	XTAL2	42	P0.1/AD1
21	XTAL1	43	P0.0/AD0
22	VSS	44	VDD

PLASTIC QUAD FLAT PACK PIN FUNCTIONS



PLASTIC QUAD FLAT PACK

Pin	Function	Pin	Function
1	P1.5	23	P2.5/A13
2	P1.6/SCL	24	P2.6/A14
3	P1.7/SDA	25	P2.7/A15
4	RST	26	PSEN
5	P3.0/RxD	27	ALE
6	VSS4	28	VSS2
7	P3.1/TxD	29	EA/Vpp
8	P3.2/INT0	30	P0.7/AD7
9	P3.3/INT1	31	P0.6/AD6
10	P3.4/T0	32	P0.5/AD5
11	P3.5/T1	33	P0.4/AD4
12	P3.6/WR	34	P0.3/AD3
13	P3.7/RD	35	P0.2/AD2
14	XTAL2	36	P0.1/AD1
15	XTAL1	37	P0.0/AD0
16	VSS1	38	VDD
17	NC	39	VSS3
18	P2.0/A8	40	P1.0
19	P2.1/A9	41	P1.1
20	P2.2/A10	42	P1.2
21	P2.3/A11	43	P1.3
22	P2.4/A12	44	P1.4

NOTES TO QFP ONLY:  
1. Due to EMC improvements, all VSS pins (6, 16, 28, 39) must be connected to VSS on the 80C652/83C652.

## CMOS single-chip 8-bit microcontroller 80C652/83C652

## ORDER INFORMATION

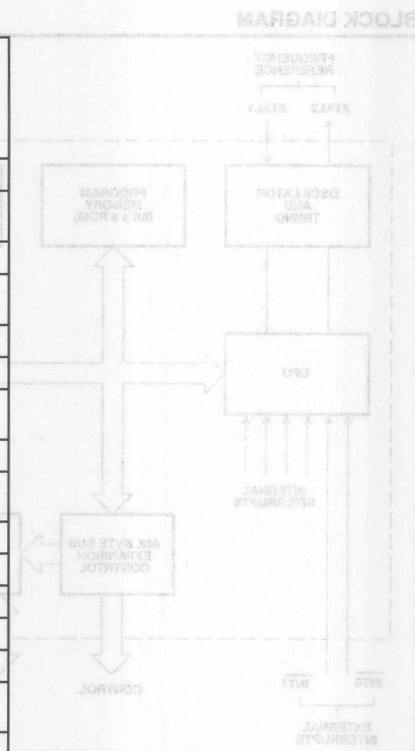
PHILIPS PART ORDER NUMBER PART MARKING			PHILIPS NORTH AMERICA PART ORDER NUMBER		TEMPERATURE RANGE (°C)  AND PACKAGE	FREQ  MHz <sup>1,2</sup>
ROMless	ROM <sup>3</sup>	Drawing Number	ROMless	ROM		
P80C652FBP	P83C652FBP/xxx	SOT129	S80C652FBPN	S83C652FBPN	0 to +70, Plastic Dual In-line Package	16
P80C652FBA	P83C652FBA/xxx	SOT187	S80C652FBAA	S83C652FBAA	0 to +70, Plastic Leaded Chip Carrier	16
P80C652FBB	P83C652FBB/xxx	SOT307-2 <sup>4</sup>	S80C652FBBB	S83C652FBBB	0 to +70, Plastic Quad Flat Pack	16
P80C652FFP	P83C652FFP/xxx	SOT129	S80C652FFPN	S83C652FFPN	-40 to +85, Plastic Dual In-line Package	16
P80C652FFA	P83C652FFA/xxx	SOT187	S80C652FFAA	S83C652FFAA	-40 to +85, Plastic Leaded Chip Carrier	16
P80C652FFB	P83C652FFB/xxx	SOT307-2 <sup>4</sup>	S80C652FFBB	S83C652FFBB	-40 to +85, Plastic Quad Flat Pack	16
P80C652FHP	P83C652FHP/xxx	SOT129	S80C652FHPN	S83C652FHPN	-40 to +125, Plastic Dual In-line Package	16
P80C652FHA	P83C652FHA/xxx	SOT187	S80C652FHAA	S83C652FHAA	-40 to +125, Plastic Leaded Chip Carrier	16
P80C652FHB	P83C652FHB/xxx	SOT307-2 <sup>4</sup>	S80C652FHBB	S83C652FHBB	-40 to +125, Plastic Quad Flat Pack	16
P80C652IBP	P83C652IBP/xxx	SOT129	S80C652IBPN	S83C652IBPN	0 to +70, Plastic Dual In-line Package	24
P80C652IBA	P83C652IBA/xxx	SOT187	S80C652IBAA	S83C652IBAA	0 to +70, Plastic Leaded Chip Carrier	24
P80C652IBB	P83C652IBB/xxx	SOT307-2 <sup>4</sup>	S80C652IBBB	S83C652IBBB	0 to +70, Plastic Quad Flat Pack	24
P80C652IFP	P83C652IFP/xxx	SOT129	S80C652IFPN	S83C652IFPN	-40 to +85, Plastic Dual In-line Package	24
P80C652IFA	P83C652IFA/xxx	SOT187	S80C652IFAA	S83C652IFAA	-40 to +85, Plastic Leaded Chip Carrier	24
P80C652IFB	P83C652IFB/xxx	SOT307-2 <sup>4</sup>	S80C652IFBB	S83C652IFBB	-40 to +85, Plastic Quad Flat Pack	24

## NOTES:

- 80C652 and 83C652 frequency range is 1.2MHz–16MHz or 1.2 to 24MHz.
- For specification of the EPROM version, see the 87C652 data sheet.
- xxx denotes the ROM code number.
- SOT311 replaced by SOT307-2.

## CMOS single-chip 8-bit microcontroller 80C652/83C652

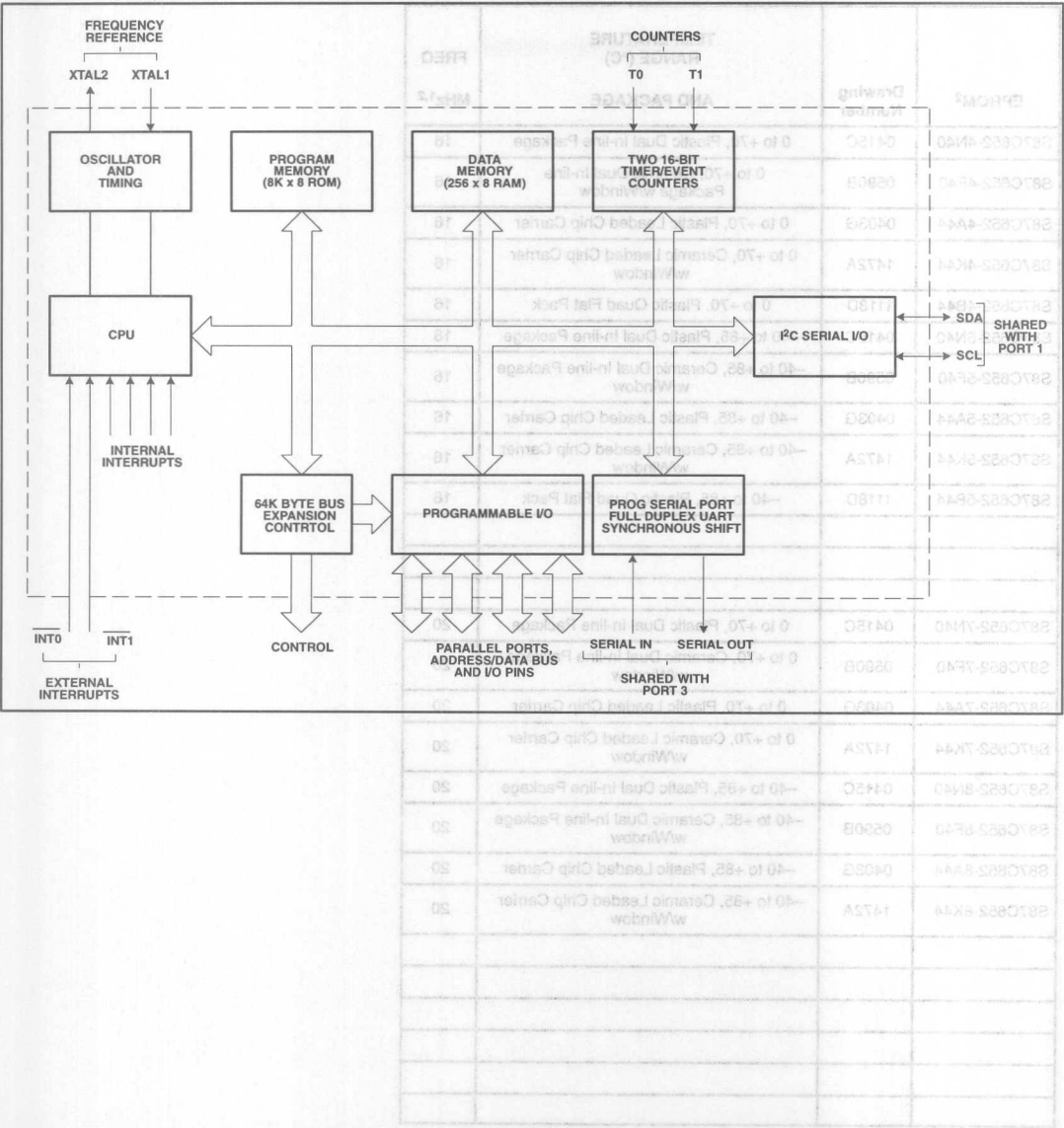
EPROM <sup>2</sup>	Drawing Number	TEMPERATURE RANGE (°C) AND PACKAGE	FREQ MHz <sup>1,2</sup>
S87C652-4N40	0415C	0 to +70, Plastic Dual In-line Package	16
S87C652-4F40	0590B	0 to +70, Ceramic Dual In-line Package w/Window	16
S87C652-4A44	0403G	0 to +70, Plastic Leaded Chip Carrier	16
S87C652-4K44	1472A	0 to +70, Ceramic Leaded Chip Carrier w/Window	16
S87C652-4B44	1118D	0 to +70, Plastic Quad Flat Pack	16
S87C652-5N40	0415C	-40 to +85, Plastic Dual In-line Package	16
S87C652-5F40	0590B	-40 to +85, Ceramic Dual In-line Package w/Window	16
S87C652-5A44	0403G	-40 to +85, Plastic Leaded Chip Carrier	16
S87C652-5K44	1472A	-40 to +85, Ceramic Leaded Chip Carrier w/Window	16
S87C652-5B44	1118D	-40 to +85, Plastic Quad Flat Pack	16
S87C652-7N40	0415C	0 to +70, Plastic Dual In-line Package	20
S87C652-7F40	0590B	0 to +70, Ceramic Dual In-line Package w/Window	20
S87C652-7A44	0403G	0 to +70, Plastic Leaded Chip Carrier	20
S87C652-7K44	1472A	0 to +70, Ceramic Leaded Chip Carrier w/Window	20
S87C652-8N40	0415C	-40 to +85, Plastic Dual In-line Package	20
S87C652-8F40	0590B	-40 to +85, Ceramic Dual In-line Package w/Window	20
S87C652-8A44	0403G	-40 to +85, Plastic Leaded Chip Carrier	20
S87C652-8K44	1472A	-40 to +85, Ceramic Leaded Chip Carrier w/Window	20





CMOS single-chip 8-bit microcontroller 80C652/83C652

BLOCK DIAGRAM



## CMOS single-chip 8-bit microcontroller 80C652/83C652

## PIN DESCRIPTIONS

MNEMONIC	PIN NUMBER			TYPE	NAME AND FUNCTION
	DIP	PLCC	QFP		
V <sub>SS</sub>	20	22	6, 16, 28, 39	I	<b>Ground:</b> 0V reference. With the QFP package all V <sub>SS</sub> pins (V <sub>SS1</sub> to V <sub>SS4</sub> ) must be connected.
V <sub>DD</sub>	40	44	38	I	<b>Power Supply:</b> This is the power supply voltage for normal, idle, and power-down operation.
P0.0–0.7	39–32	43–36	37–30	I/O	<b>Port 0:</b> Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s.
P1.0–P1.7	1–8	2–9	40–44, 1–3	I/O	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pull-ups, except P1.6 and P1.7 which are open drain. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Alternate functions include: <b>SCL:</b> I <sup>2</sup> C-bus serial port clock line. <b>SDA:</b> I <sup>2</sup> C-bus serial port data line.
P1.6	7	8	2	I/O	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register.
P1.7	8	9	3	I/O	
P2.0–P2.7	21–28	24–31	18–25	I/O	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 3 also serves the special features of the 80C51 family, as listed below: <b>RxD (P3.0):</b> Serial input port <b>TxD (P3.1):</b> Serial output port <b>INT0 (P3.2):</b> External interrupt <b>INT1 (P3.3):</b> External interrupt <b>T0 (P3.4):</b> Timer 0 external input <b>T1 (P3.5):</b> Timer 1 external input <b>WR (P3.6):</b> External data memory write strobe <b>RD (P3.7):</b> External data memory read strobe
P3.0–P3.7	10–17	11, 13–19	5, 7–13	I/O	
RST	10	11	5	I	<b>Reset:</b> A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to V <sub>SS</sub> permits a power-on reset using only an external capacitor to V <sub>DD</sub> .
	11	13	7	O	
	12	14	8	I	
	13	15	9	I	
	14	16	10	I	
	15	17	11	I	
	16	18	12	O	
	17	19	13	O	
ALE	30	33	27	I/O	<b>Address Latch Enable:</b> Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency. Note that one ALE pulse is skipped during each access to external data memory.
PSEN	29	32	26	O	<b>Program Store Enable:</b> Read strobe to external program memory via Port 0 and Port 2. It is activated twice each machine cycle during fetches from the external program memory. When executing out of external program memory two activations of PSEN are skipped during each access to external data memory. PSEN is not activated (remains HIGH) during no fetches from external program memory. PSEN can sink/source 8 LSTTL inputs and can drive CMOS inputs without external pull-ups.
EA	31	35	29	I	<b>External Access:</b> If during a RESET, EA is held at TTL, level HIGH, the CPU executes out of the internal program memory ROM provided the Program Counter is less than 8192. If during a RESET, EA is held a TTL LOW level, the CPU executes out of external program memory. EA is not allowed to float.
XTAL1	19	21	15	I	<b>Crystal 1:</b> Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	20	14	O	<b>Crystal 2:</b> Output from the inverting oscillator amplifier.

## NOTE:

To avoid "latch-up" effect at power-on, the voltage on any pin at any time must not be higher than V<sub>DD</sub> + 0.5V or V<sub>SS</sub> – 0.5V, respectively.

## CMOS single-chip 8-bit microcontroller 80C652/83C652

Table 1. 8XC652/654 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB				LSB				
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
DPTR:	Data pointer (2 bytes)										
DPH	Data pointer high	83H									00H
DPL	Data pointer low	82H									00H
			AF	AE	AD	AC	AB	AA	A9	A8	
IE*#	Interrupt enable	A8H	EA		ES1	ES0	ET1	EX1	ET0	EX0	0x000000B
			BF	BE	BD	BC	BB	BA	B9	B8	
IP*#	Interrupt priority	B8H	—		PS1	PS0	PT1	PX1	PT0	PX0	xx000000B
			87	86	85	84	83	82	81	80	
P0*	Port 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
			97	96	95	94	93	92	91	90	
P1*#	Port 1	90H	SDA	SCL							FFH
			A7	A6	A5	A4	A3	A2	A1	A0	
P2*	Port 2	A0H	A15	A14	A13	A12	A11	A10	A9	A8	FFH
			B7	B6	B5	B4	B3	B2	B1	B0	
P3*	Port 3	B0H	RD	WR	T1	T0	INT1	INT0	TXD	RXD	FFH
PCON	Power control	87H	SMOD	—	—	—	GF1	GF0	PD	IDL	0xxx0000B
			9F	9E	9D	9C	9B	9A	99	98	
S0CON*#	Serial 0 port control	98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00H
S0BUF#	Serial 0 data buffer	99H									xxxxxxxxB
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program status word	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	00H
S1DAT#	Serial 1 data	DAH									00H
SP	Stack pointer	81H									07H
S1ADR#	Serial 1 address	DBH	SLAVE ADDRESS							GC	00H
S1STA#	Serial 1 status	D9H	SC4	SC3	SC2	SC1	SC0	0	0	0	F8H
			DF	DE	DD	DC	DB	DA	D9	D8	
S1CON*#	Serial 1 control	D8H	CR2	ENS1	STA	STO	SI	AA	CR1	CR0	00000000B
			8F	8E	8D	8C	8B	8A	89	88	
TCON*	Timer control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
TH1	Timer high 1	8DH									00H
TH0	Timer high 0	8CH									00H
TL1	Timer low 1	8BH									00H
TH0	Timer low 0	8AH									00H
TMOD	Timer mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.

## CMOS single-chip 8-bit microcontroller 80C652/83C652

**ROM CODE PROTECTION  
(83C652)**

The 8XC652 has an additional security feature. ROM code protection may be selected by setting a mask-programmable security bit (i.e., user dependent). This feature may be requested during ROM code submission. When selected, the ROM code is protected and cannot be read out at any time by any test mode or by any instruction in the external program memory space.

The MOV<sub>C</sub> instructions are the only instructions that have access to program code in the internal or external program memory. The EA input is latched during RESET and is "don't care" after RESET (also if the security bit is not set). This implementation prevents reading internal program code by switching from external program memory to internal program memory during a MOV<sub>C</sub> instruction or any other instruction that uses immediate data.

**OSCILLATOR  
CHARACTERISTICS**

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator, as shown in the Logic Symbol, page 916.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

**Reset**

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. To insure a good power-on reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few milliseconds) plus two machine cycles. At power-on, the voltage on V<sub>DD</sub> and RST must come up at the same time for a proper start-up.

**Idle Mode**

In the idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle mode can be terminated either by any

enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

**Power-Down Mode**

In the power-down mode, the oscillator is stopped and the instruction to invoke power-down is the last instruction executed. Only the contents of the on-chip RAM are preserved. A hardware reset is the only way to terminate the power-down mode. The control bits for the reduced power modes are in the special function register PCON. Table 2 shows the state of the I/O ports during low current operating modes.

**I<sup>2</sup>C Serial Communication—SIO1**

The I<sup>2</sup>C serial port is identical to the I<sup>2</sup>C serial port on the 8XC552. The operation of this subsystem is described in detail in the 8XC552 section of this manual.

Note that in both the 8XC652/4 and the 8XC552 the I<sup>2</sup>C pins are alternate functions to port pins P1.6 and P1.7. Because of this, P1.6 and P1.7 on these parts do not have a pull-up structure as found on the 80C51. Therefore P1.6 and P1.7 have open drain outputs on the 8XC652/4.

**Table 2. External Pin Status During Idle and Power-Down Mode**

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

**Serial Control Register (S1CON) – See Table 3**

S1CON (D8H)	CR2	ENS1	STA	STO	SI	AA	CR1	CR0
-------------	-----	------	-----	-----	----	----	-----	-----

Bits CR0, CR1 and CR2 determine the serial clock frequency that is generated in the master mode of operation.

**Table 3. Serial Clock Rates**

CR2	CR1	CR0	BIT FREQUENCY (kHz) AT f <sub>osc</sub>				f <sub>osc</sub> DIVIDED BY
			6MHz	12MHz	16MHz	24MHz	
0	0	0	23	47	62.5	94	256
0	0	1	27	54	71	107 <sup>1</sup>	224
0	1	0	31.25	62.5	83.3	125 <sup>1</sup>	192
0	1	1	37	75	100	150 <sup>1</sup>	160
1	0	0	6.25	12.5	17	25	960
1	0	1	50	100	133 <sup>1</sup>	200 <sup>1</sup>	120
1	1	0	100	200 <sup>1</sup>	267 <sup>1</sup>	400 <sup>1</sup>	60
1	1	1	0.24 < 62.5 0 to 255	0.49 < 62.5 0 to 254	0.65 < 55.6 0 to 253	0.98 < 50.0 0 to 251	96 × (256 – (reload value Timer 1)) reload value range Timer 1 (in mode 2)

**NOTES:**

1. These frequencies exceed the upper limit of 100kHz of the I<sup>2</sup>C-bus specification and cannot be used in an I<sup>2</sup>C-bus application.

## CMOS single-chip 8-bit microcontroller 80C652/83C652

**ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>**

PARAMETER	RATING	UNIT
Storage temperature range	-65 to +150	°C
Voltage on any other pin to V <sub>SS</sub>	-0.5 to +6.5	V
Input, output current on any single pin	±5	mA
Power dissipation (based on package limitations, not device power consumption)	1	W

**NOTES:**

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
2. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
3. Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to  $V_{EE}$  unless otherwise noted.

## DEVICE SPECIFICATIONS

TYPE	SUPPLY VOLTAGE (V)		FREQUENCY (MHz)		TEMPERATURE RANGE
	MIN.	MAX.	MIN.	MAX.	(°C)
P8XC652FBx	4.0	6.0	1.2	16	0 to +70
P8XC652FFx	4.0	6.0	1.2	16	-40 to +85
P8XC652FHx	4.5	5.5	1.2	16	-40 to +125
P8XC652IBx	4.5	5.5	1.2	24	0 to +70
P83X652IFx	4.5	5.5	1.2	24	-40 to +85

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Run	External	1	1	Float	Float	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data



## CMOS single-chip 8-bit microcontroller 80C652/83C652

## DC ELECTRICAL CHARACTERISTICS

 $V_{SS} = 0V$ 

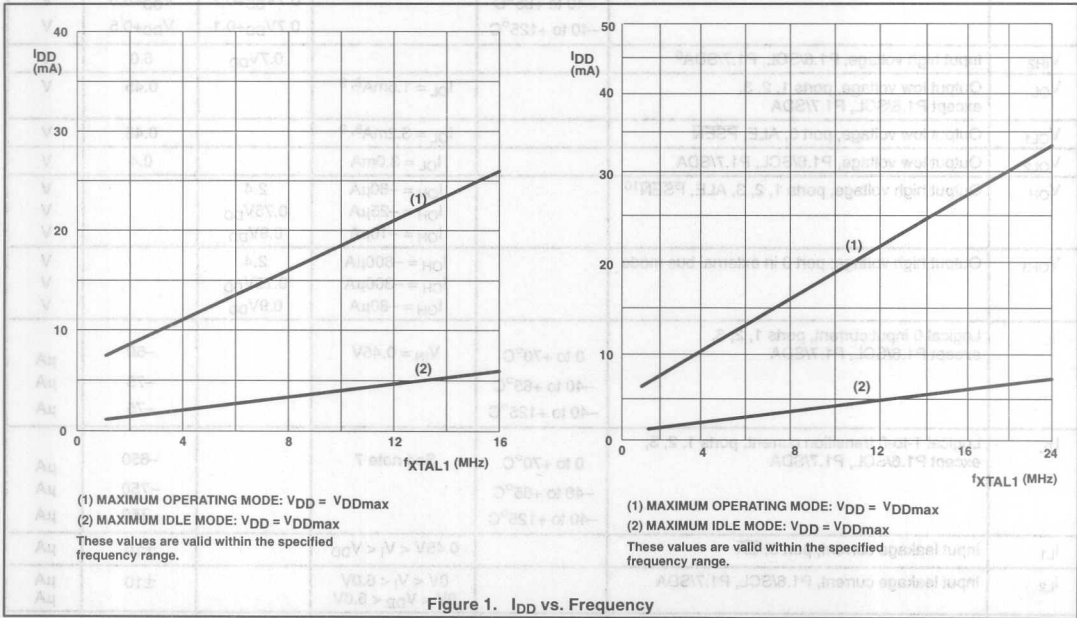
SYMBOL	PARAMETER	PART TYPE	TEST CONDITIONS	LIMITS		UNIT
				MIN.	MAX.	
$V_{IL}$	Input low voltage, except EA, P1.6/SCL, P1.7/SDA	0 to +70°C		-0.5	$0.2V_{DD}-0.1$	V
		-40 to +85°C		-0.5	$0.2V_{DD}-0.15$	V
		-40 to +125°C		-0.5	$0.2V_{DD}-0.25$	V
$V_{IL1}$	Input low voltage to EA	0 to +70°C		-0.5	$0.2V_{DD}-0.3$	V
		-40 to +85°C		-0.5	$0.2V_{DD}-0.35$	V
		-40 to +125°C		-0.5	$0.2V_{DD}-0.45$	V
$V_{IL2}$	Input low voltage to P1.6/SCL, P1.7/SDA <sup>6</sup>			-0.5	$0.3V_{DD}$	V
$V_{IH}$	Input high voltage, except XTAL1, RST, P1.6/SCL, P1.7/SDA	0 to +70°C		$0.2V_{DD}+0.9$	$V_{DD}+0.5$	V
		-40 to +85°C		$0.2V_{DD}+1.0$	$V_{DD}+0.5$	V
		-40 to +125°C		$0.2V_{DD}+1.0$	$V_{DD}+0.5$	V
$V_{IH1}$	Input high voltage, XTAL1, RST	0 to +70°C		$0.7V_{DD}$	$V_{DD}+0.5$	V
		-40 to +85°C		$0.7V_{DD}+0.1$	$V_{DD}+0.5$	V
		-40 to +125°C		$0.7V_{DD}+0.1$	$V_{DD}+0.5$	V
$V_{IH2}$	Input high voltage, P1.6/SCL, P1.7/SDA <sup>6</sup>			$0.7V_{DD}$	6.0	V
$V_{OL}$	Output low voltage, ports 1, 2, 3, except P1.6/SCL, P1.7/SDA		$I_{OL} = 1.6mA^{8,9}$		0.45	V
$V_{OL1}$	Output low voltage, port 0, ALE, PSEN		$I_{OL} = 3.2mA^{8,9}$		0.45	V
$V_{OL2}$	Output low voltage, P1.6/SCL, P1.7/SDA		$I_{OL} = 3.0mA$		0.4	V
$V_{OH}$	Output high voltage, ports 1, 2, 3, ALE, PSEN <sup>10</sup>		$I_{OH} = -60\mu A$	2.4		V
			$I_{OH} = -25\mu A$	$0.75V_{DD}$		V
			$I_{OH} = -10\mu A$	$0.9V_{DD}$		V
$V_{OH1}$	Output high voltage; port 0 in external bus mode		$I_{OH} = -800\mu A$	2.4		V
			$I_{OH} = -300\mu A$	$0.75V_{DD}$		V
			$I_{OH} = -80\mu A$	$0.9V_{DD}$		V
$I_{IL}$	Logical 0 input current, ports 1, 2, 3, except P1.6/SCL, P1.7/SDA	0 to +70°C	$V_{IN} = 0.45V$		-50	$\mu A$
		-40 to +85°C			-75	$\mu A$
		-40 to +125°C			-75	$\mu A$
$I_{TL}$	Logical 1-to-0 transition current, ports 1, 2, 3, except P1.6/SCL, P1.7/SDA	0 to +70°C	See note 7		-650	$\mu A$
		-40 to +85°C			-750	$\mu A$
		-40 to +125°C			-750	$\mu A$
$I_{L1}$	Input leakage current, port 0, EA		$0.45V < V_I < V_{DD}$		$\pm 10$	$\mu A$
$I_{L2}$	Input leakage current, P1.6/SCL, P1.7/SDA		$0V < V_I < 6.0V$ $0V < V_{DD} < 6.0V$		$\pm 10$	$\mu A$
$I_{DD}$	Power supply current: Active mode @ 16MHz <sup>2, 11</sup> Active mode @ 24MHz <sup>2, 11</sup> Idle mode @ 16MHz <sup>3, 11</sup> Idle mode @ 24MHz <sup>3, 11</sup> Power down mode <sup>4, 5</sup> Power down mode <sup>4, 5</sup>		See note 1 $V_{DD}=6.0V$ $V_{DD}=5.5V$		26.5	mA
					33.8	mA
					6	mA
					7	mA
					50	$\mu A$
					100	$\mu A$
$R_{RST}$	Internal reset pull-down resistor			50	150	k $\Omega$
$C_{IO}$	Pin capacitance		Freq.=1MHz		10	pF

NOTES ON NEXT PAGE.

CMOS single-chip 8-bit microcontroller 80C652/83C652

NOTES FOR DC ELECTRICAL CHARACTERISTICS:

1. See Figures 9 through 11 for  $I_{DD}$  test conditions.
2. The operating supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 5\text{ns}$ ;  $V_{IL} = V_{SS} + 0.5\text{V}$ ;  $V_{IH} = V_{DD} - 0.5\text{V}$ ; XTAL2 not connected;  $\overline{\text{EA}} = \text{RST} = \text{Port } 0 = \text{P1.6} = \text{P1.7} = V_{DD}$ . See Figure 9.
3. The idle mode supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 5\text{ns}$ ;  $V_{IL} = V_{SS} + 0.5\text{V}$ ;  $V_{IH} = V_{DD} - 0.5\text{V}$ ; XTAL2 not connected; Port 0 = P1.6 = P1.7 =  $V_{DD}$ ;  $\overline{\text{EA}} = \text{RST} = V_{SS}$ . See Figure 10.
4. The power-down current is measured with all output pins disconnected; XTAL2 not connected; Port 0 = P1.6 = P1.7 =  $V_{DD}$ ;  $\overline{\text{EA}} = \text{RST} = V_{SS}$ . See Figure 11.
5.  $2\text{V} \leq V_{PD} \leq V_{DD\text{max}}$ .
6. The input threshold voltage of P1.6 and P1.7 (SIO1) meets the  $I^2\text{C}$  specification, so an input voltage below  $0.3V_{DD}$  will be recognized as a logic 0 while an input voltage above  $0.7V_{DD}$  will be recognized as a logic 1.
7. Pins of ports 1, 2, and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when  $V_{IN}$  is approximately 2V.
8. Capacitive loading on ports 0 and 2 may cause spurious noise to be superimposed on the  $V_{OL}$ s of ALE and ports 1 and 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading  $> 100\text{pF}$ ), the noise pulse on the ALE pin may exceed  $0.8\text{V}$ . In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.
9. Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows: Maximum  $I_{OL} = 10\text{mA}$  per port pin; Maximum  $I_{OL} = 26\text{mA}$  total for Port 0; Maximum  $I_{OL} = 15\text{mA}$  total for Ports 1, 2, and 3; Maximum  $I_{OL} = 71\text{mA}$  total for all output pins. If  $I_{OL}$  exceeds the test conditions,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
10. Capacitive loading on ports 0 and 2 may cause the  $V_{OH}$  on ALE and PSEN to momentarily fall below the  $0.9V_{DD}$  specification when the address bits are stabilizing.
11.  $I_{DD\text{max}}$  for other frequencies can be derived from Figure 1, where  $\text{FREQ}$  is the external oscillator frequency in MHz.  $I_{DD\text{max}}$  is given in mA.



## CMOS single-chip 8-bit microcontroller 80C652/83C652

AC ELECTRICAL CHARACTERISTICS<sup>1, 2</sup> (16 MHz type)

SYMBOL	FIGURE	PARAMETER	16MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
$f_{\text{CLCL}}$	2	Oscillator frequency			1.2	16	MHz
$t_{\text{LHLL}}$	2	ALE pulse width	85		$2t_{\text{CLCL}}-40$		ns
$t_{\text{AVLL}}$	2	Address valid to ALE low	8		$t_{\text{CLCL}}-55$		ns
$t_{\text{LLAX}}$	2	Address hold after ALE low	28		$t_{\text{CLCL}}-35$		ns
$t_{\text{LLIV}}$	2	ALE low to valid instruction in		150		$4t_{\text{CLCL}}-100$	ns
$t_{\text{LLPL}}$	2	ALE low to PSEN low	23		$t_{\text{CLCL}}-40$		ns
$t_{\text{PLPH}}$	2	PSEN pulse width	143		$3t_{\text{CLCL}}-45$		ns
$t_{\text{PLIV}}$	2	PSEN low to valid instruction in		83		$3t_{\text{CLCL}}-105$	ns
$t_{\text{PXIX}}$	2	Input instruction hold after PSEN	0		0		ns
$t_{\text{PXIZ}}$	2	Input instruction float after PSEN		38		$t_{\text{CLCL}}-25$	ns
$t_{\text{AVIV}}$	2	Address to valid instruction in		208		$5t_{\text{CLCL}}-105$	ns
$t_{\text{PLAZ}}$	2	PSEN low to address float		10		10	ns
<b>Data Memory</b>							
$t_{\text{RLRH}}$	3, 4	RD pulse width	275		$6t_{\text{CLCL}}-100$		ns
$t_{\text{WLWH}}$	3, 4	WR pulse width	275		$6t_{\text{CLCL}}-100$		ns
$t_{\text{RLDV}}$	3, 4	RD low to valid data in		148		$5t_{\text{CLCL}}-165$	ns
$t_{\text{RHDX}}$	3, 4	Data hold after RD	0		0		ns
$t_{\text{RHDZ}}$	3, 4	Data float after RD		55		$2t_{\text{CLCL}}-70$	ns
$t_{\text{LLDV}}$	3, 4	ALE low to valid data in		350		$8t_{\text{CLCL}}-150$	ns
$t_{\text{AVDV}}$	3, 4	Address to valid data in		398		$9t_{\text{CLCL}}-165$	ns
$t_{\text{LLWL}}$	3, 4	ALE low to RD or WR low	138	238	$3t_{\text{CLCL}}-50$	$3t_{\text{CLCL}}+50$	ns
$t_{\text{AVWL}}$	3, 4	Address valid to WR low or RD low	120		$4t_{\text{CLCL}}-120$		ns
$t_{\text{QVWX}}$	3, 4	Data valid to WR transition	3		$t_{\text{CLCL}}-60$		ns
$t_{\text{DW}}$	3, 4	Data setup time before WR	288		$7t_{\text{CLCL}}-150$		ns
$t_{\text{WHQX}}$	3, 4	Data hold after WR	13		$t_{\text{CLCL}}-50$		ns
$t_{\text{RLAZ}}$	3, 4	RD low to address float		0		0	ns
$t_{\text{WHLH}}$	3, 4	RD or WR high to ALE high	23	103	$t_{\text{CLCL}}-40$	$t_{\text{CLCL}}+40$	ns
<b>Shift Register</b>							
$t_{\text{XLXL}}$	5	Serial port clock cycle time <sup>3</sup>	0.75		$12t_{\text{CLCL}}$		μs
$t_{\text{QVXH}}$	5	Output data setup to clock rising edge <sup>3</sup>	492		$10t_{\text{CLCL}}-133$		ns
$t_{\text{XHQX}}$	5	Output data hold after clock rising edge <sup>3</sup>	80		$2t_{\text{CLCL}}-117$		ns
$t_{\text{XHDX}}$	5	Input data hold after clock rising edge <sup>3</sup>	0		0		ns
$t_{\text{XHDV}}$	5	Clock rising edge to input data valid <sup>3</sup>		492		$10t_{\text{CLCL}}-133$	ns
<b>External Clock</b>							
$t_{\text{CHCX}}$	6	High time <sup>3</sup>	20		20	$t_{\text{CLCL}} - t_{\text{CLCX}}$	ns
$t_{\text{CLCX}}$	6	Low time <sup>3</sup>	20		20	$t_{\text{CLCL}} - t_{\text{CHCX}}$	ns
$t_{\text{CLCH}}$	6	Rise time <sup>3</sup>		20		20	ns
$t_{\text{CHCL}}$	6	Fall time <sup>3</sup>		20		20	ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- These values are characterized but not 100% production tested.

AC ELECTRICAL CHARACTERISTICS<sup>1, 2</sup> (24 MHz type)

SYMBOL	FIGURE	PARAMETER	24MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
$1/t_{CLCL}$	2	Oscillator frequency			1.2	24	MHz
$t_{LHLL}$	2	ALE pulse width	43		$2t_{CLCL}-40$		ns
$t_{AVLL}$	2	Address valid to ALE low	17		$t_{CLCL}-25$		ns
$t_{LLAX}$	2	Address hold after ALE low	17		$t_{CLCL}-25$		ns
$t_{LLIV}$	2	ALE low to valid instruction in		102		$4t_{CLCL}-65$	ns
$t_{LLPL}$	2	ALE low to PSEN low	17		$t_{CLCL}-25$		ns
$t_{PLPH}$	2	PSEN pulse width	80		$3t_{CLCL}-45$		ns
$t_{PLIV}$	2	PSEN low to valid instruction in		65		$3t_{CLCL}-60$	ns
$t_{PXIX}$	2	Input instruction hold after PSEN	10		0		ns
$t_{PIXZ}$	2	Input instruction float after PSEN		17		$t_{CLCL}-25$	ns
$t_{AVIV}$	2	Address to valid instruction in		128		$5t_{CLCL}-80$	ns
$t_{PLAZ}$	2	PSEN low to address float		10		10	ns
<b>Data Memory</b>							
$t_{RLRH}$	3, 4	RD pulse width	150		$6t_{CLCL}-100$		ns
$t_{WLWH}$	3, 4	WR pulse width	150		$6t_{CLCL}-100$		ns
$t_{RLDV}$	3, 4	RD low to valid data in		118		$5t_{CLCL}-90$	ns
$t_{RHDX}$	3, 4	Data hold after RD	0		0		ns
$t_{RHDZ}$	3, 4	Data float after RD		55		$2t_{CLCL}-28$	ns
$t_{LLDV}$	3, 4	ALE low to valid data in		180		$8t_{CLCL}-150$	ns
$t_{AVDV}$	3, 4	Address to valid data in		210		$9t_{CLCL}-165$	ns
$t_{LLWL}$	3, 4	ALE low to RD or WR low	75	175	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
$t_{AVWL}$	3, 4	Address valid to WR low or RD low	92		$4t_{CLCL}-75$		ns
$t_{QVWX}$	3, 4	Data valid to WR transition	12		$t_{CLCL}-30$		ns
$t_{DW}$	3, 4	Data setup time before WR	162		$7t_{CLCL}-130$		ns
$t_{WHQX}$	3, 4	Data hold after WR	17		$t_{CLCL}-25$		ns
$t_{RLAZ}$	3, 4	RD low to address float		0		0	ns
$t_{WHLH}$	3, 4	RD or WR high to ALE high	17	67	$t_{CLCL}-25$	$t_{CLCL}+25$	ns
<b>Shift Register</b>							
$t_{XLXL}$	5	Serial port clock cycle time <sup>3</sup>	0.5		$12t_{CLCL}$		μs
$t_{QVXH}$	5	Output data setup to clock rising edge <sup>3</sup>	283		$10t_{CLCL}-133$		ns
$t_{XHDX}$	5	Output data hold after clock rising edge <sup>3</sup>	23		$2t_{CLCL}-60$		ns
$t_{XHDX}$	5	Input data hold after clock rising edge <sup>3</sup>	0		0		ns
$t_{XHDV}$	5	Clock rising edge to input data valid <sup>3</sup>		283		$10t_{CLCL}-133$	ns
<b>External Clock</b>							
$t_{CHCX}$	6	High time <sup>3</sup>	17		17	$t_{CLCL}-t_{CLCX}$	ns
$t_{CLCX}$	6	Low time <sup>3</sup>	17		17	$t_{CLCL}-t_{CHCX}$	ns
$t_{CLCH}$	6	Rise time <sup>3</sup>		5		5	ns
$t_{CHCL}$	6	Fall time <sup>3</sup>		5		5	ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- These values are characterized but not 100% production tested.

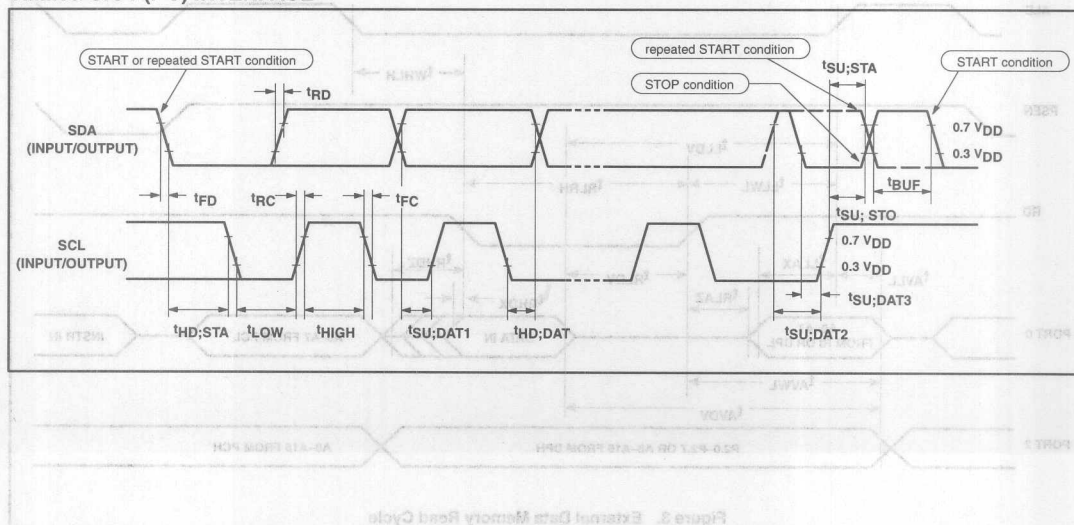
## CMOS single-chip 8-bit microcontroller 80C652/83C652

AC ELECTRICAL CHARACTERISTICS – I<sup>2</sup>C INTERFACE

SYMBOL	PARAMETER	INPUT	OUTPUT
<b>SCL TIMING CHARACTERISTICS</b>			
$t_{HD; STA}$	START condition hold time	$\geq 14 t_{CLCL}$	$> 4.0\mu s^1$
$t_{LOW}$	SCL LOW time	$\geq 16 t_{CLCL}$	$> 4.7\mu s^1$
$t_{HIGH}$	SCL HIGH time	$\geq 14 t_{CLCL}$	$> 4.0\mu s^1$
$t_{RC}$	SCL rise time	$\leq 1\mu s$	$\leq 2$
$t_{FC}$	SCL fall time	$\leq 0.3\mu s$	$< 0.3\mu s^3$
<b>SDA TIMING CHARACTERISTICS</b>			
$t_{SU; DAT1}$	Data set-up time	$\geq 250ns$	$> 20 t_{CLCL} - t_{RD}$
$t_{SU; DAT2}$	SDA set-up time (before rep. START cond.)	$\geq 250ns$	$> 1\mu s^1$
$t_{SU; DAT3}$	SDA set-up time (before STOP cond.)	$\geq 250ns$	$> 8 t_{CLCL}$
$t_{HD; DAT}$	Data hold time	$\geq 0ns$	$> 8 t_{CLCL} - t_{FC}$
$t_{SU; STA}$	Repeated START set-up time	$\geq 14 t_{CLCL}$	$> 4.7\mu s^1$
$t_{SU; STO}$	STOP condition set-up time	$\geq 14 t_{CLCL}$	$> 4.0\mu s^1$
$t_{BUF}$	Bus free time	$\geq 14 t_{CLCL}$	$> 4.7\mu s^1$
$t_{RD}$	SDA rise time	$\leq 1\mu s$	$-^2$
$t_{FD}$	SDA fall time	$\leq 0.3\mu s$	$< 0.3\mu s^3$

## NOTES:

- At 100 kbit/s. At other bit rates this value is inversely proportional to the bit-rate of 100 kbit/s.
- Determined by the external bus-line capacitance and the external bus-line pull-resistor, this must be  $< 1\mu s$ .
- Spikes on the SDA and SCL lines with a duration of less than  $3 t_{CLCL}$  will be filtered out. Maximum capacitance on bus-lines SDA and SCL = 400pF.
- $t_{CLCL} = 1/f_{OSC}$  = one oscillator clock period at pin XTAL1. For 63ns (42ns)  $< t_{CLCL} < 285ns$  (16MHz (24MHz)  $> f_{OSC} > 3.5MHz$ ) the SI01 interface meets the I<sup>2</sup>C-bus specification for bit-rates up to 100 kbit/s.

TIMING SI01 (I<sup>2</sup>C) INTERFACE



# CMOS single-chip 8-bit microcontroller

80C652/83C652

## EXPLANATION OF THE AC SYMBOLS

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

- A – Address
- C – Clock
- D – Input data
- H – Logic level high
- I – Instruction (program memory contents)
- L – Logic level low, or ALE
- P – PSEN

- Q – Output data
- R – RD signal
- t – Time
- V – Valid
- W – WR signal
- X – No longer a valid logic level
- Z – Float

Examples:  $t_{AVLL}$  = Time for address valid to ALE low.  
 $t_{LLPL}$  = Time for ALE low to PSEN low.

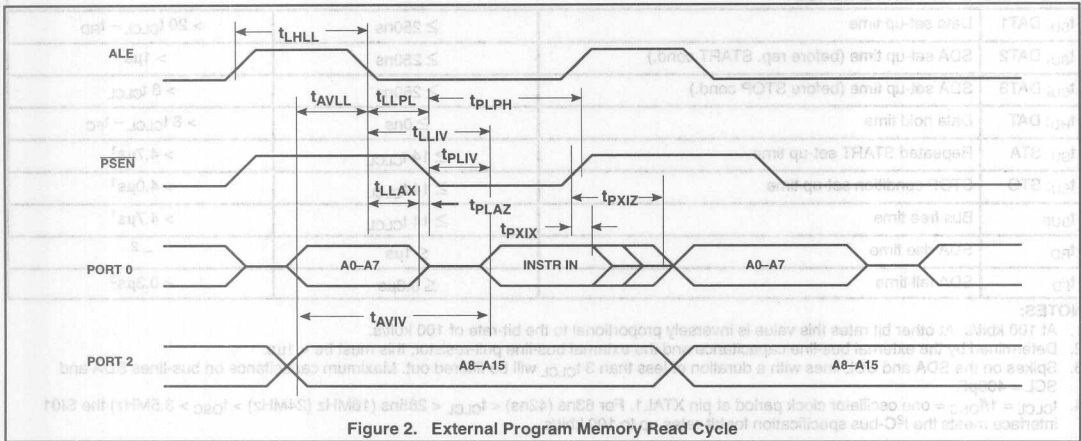


Figure 2. External Program Memory Read Cycle

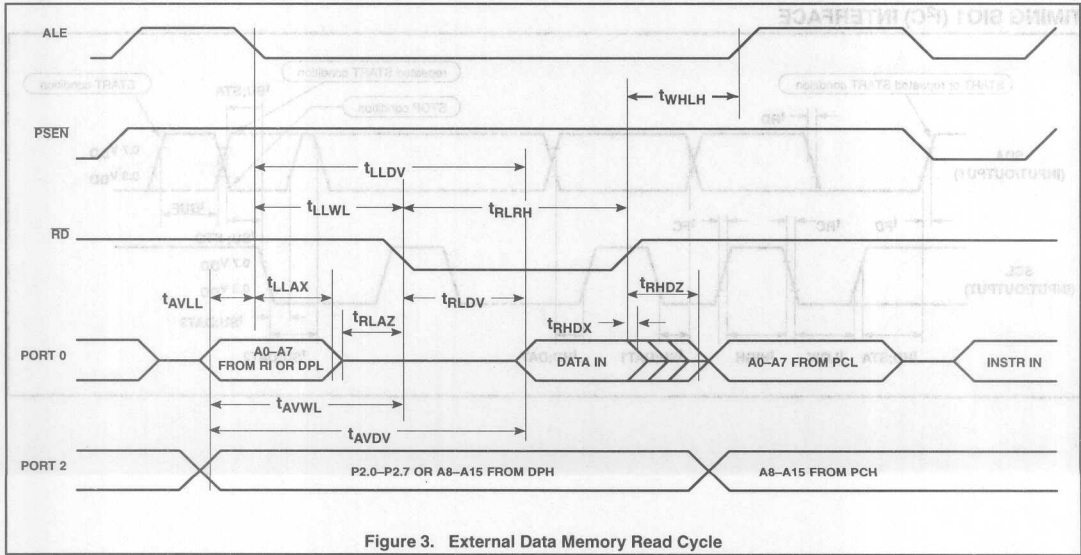
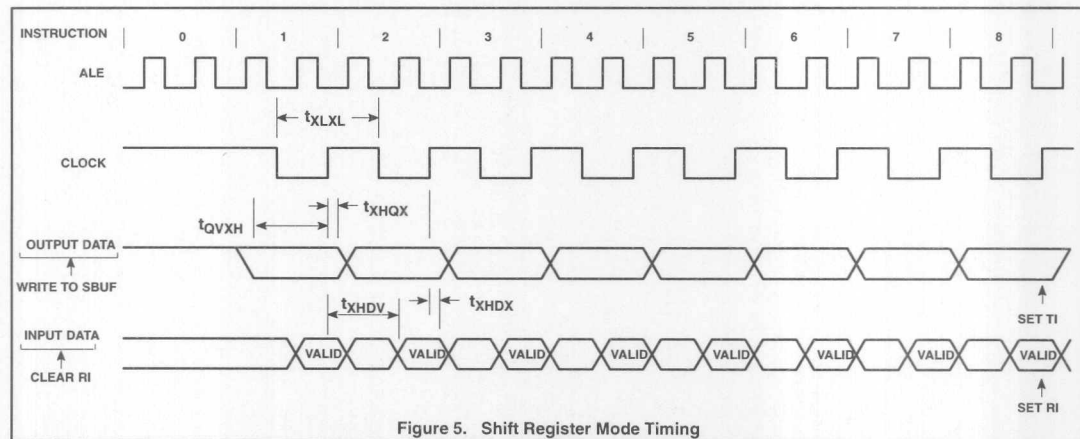
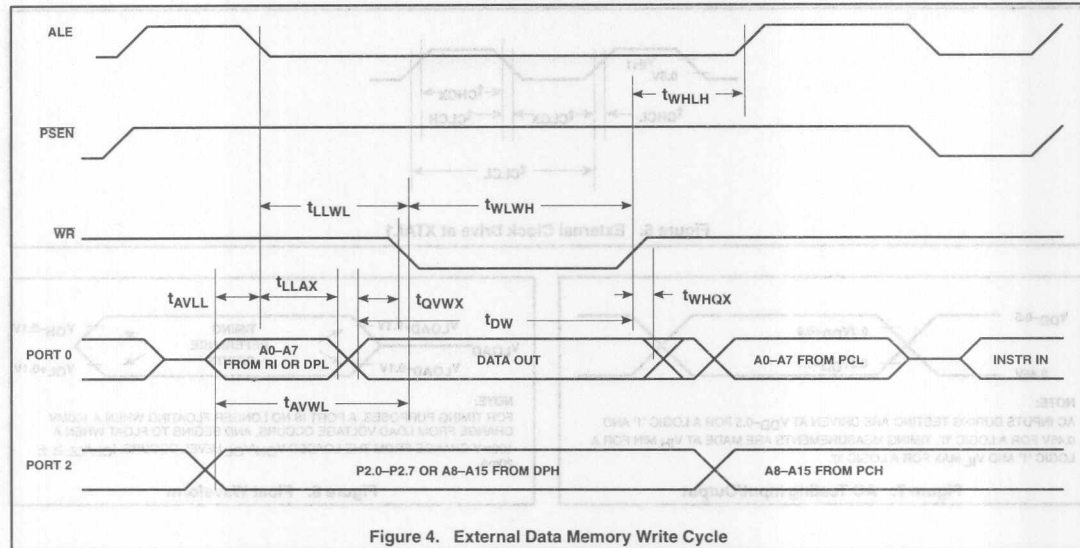
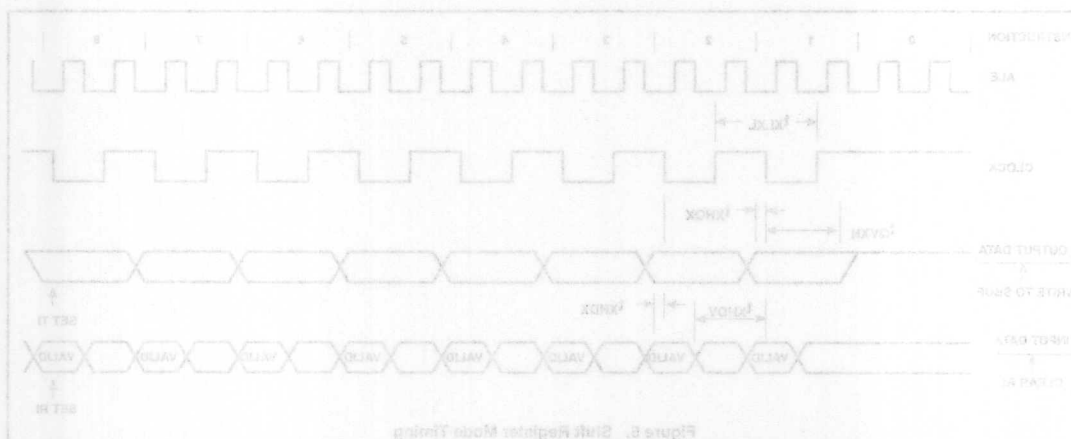
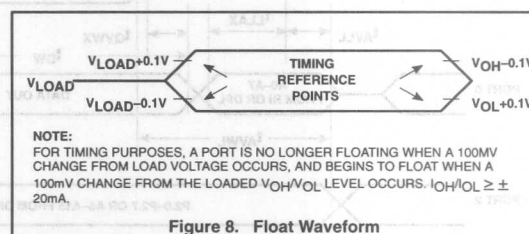
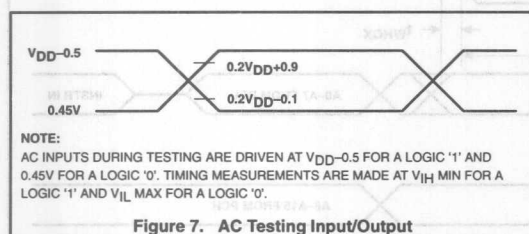
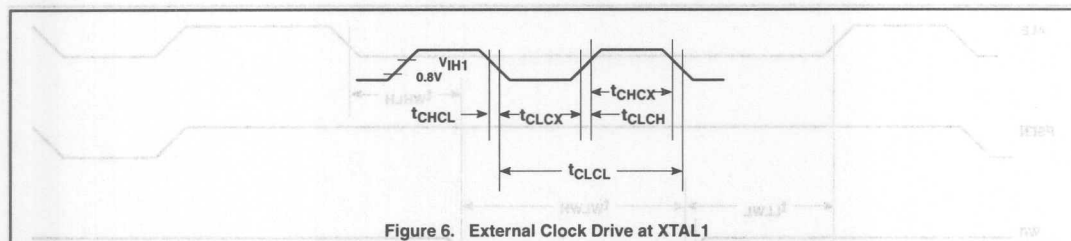


Figure 3. External Data Memory Read Cycle

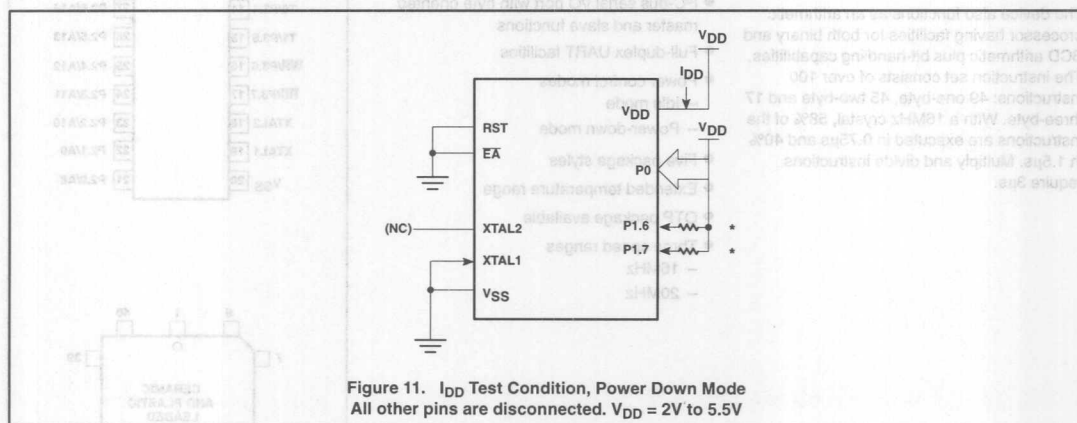
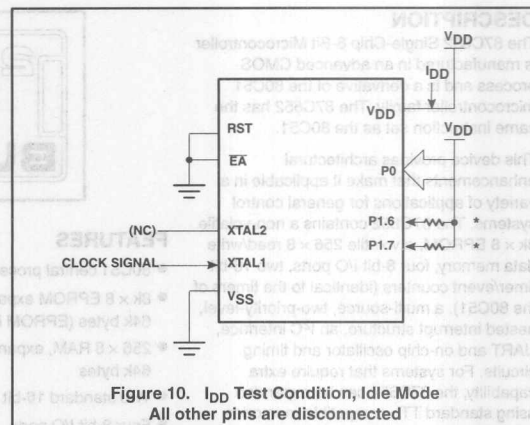
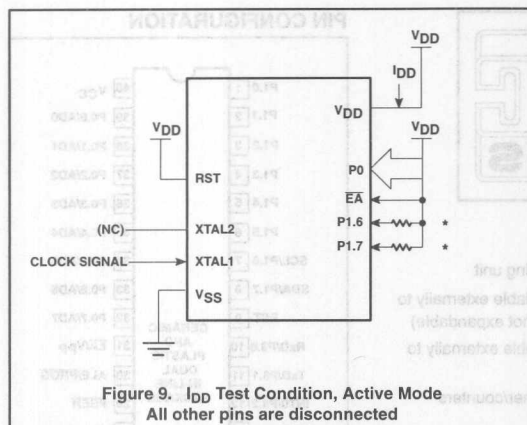
## CMOS single-chip 8-bit microcontroller 80C652/83C652



## CMOS single-chip 8-bit microcontroller 80C652/83C652



## CMOS single-chip 8-bit microcontroller 80C652/83C652

**NOTE:**

- \* Ports 1.6 and 1.7 should be connected to  $V_{CC}$  through resistors of sufficiently high value such that the sink current into these pins does not exceed the  $I_{OL1}$  specification.



Purchase of Philips' I<sup>2</sup>C components conveys a license under the Philips' I<sup>2</sup>C patent to use the components in the I<sup>2</sup>C-system provided the system conforms to the I<sup>2</sup>C specifications defined by Philips.

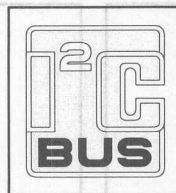
# CMOS single-chip 8-bit microcontroller 87C652

## DESCRIPTION

The 87C652 Single-Chip 8-Bit Microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 87C652 has the same instruction set as the 80C51.

This device provides architectural enhancements that make it applicable in a variety of applications for general control systems. The 87C652 contains a non-volatile 8k × 8 EPROM, a volatile 256 × 8 read/write data memory, four 8-bit I/O ports, two 16-bit timer/event counters (identical to the timers of the 80C51), a multi-source, two-priority-level, nested interrupt structure, an I<sup>2</sup>C interface, UART and on-chip oscillator and timing circuits. For systems that require extra capability, the 87C652 can be expanded using standard TTL compatible memories and logic.

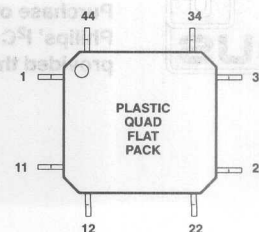
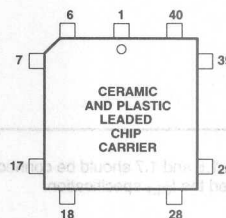
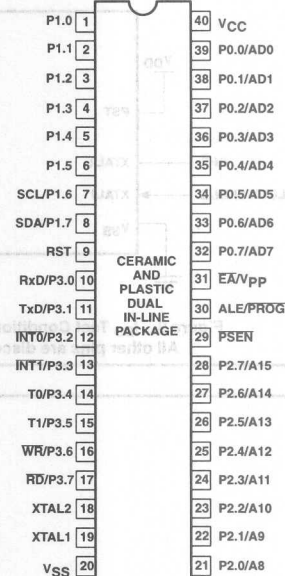
The device also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of over 100 instructions: 49 one-byte, 45 two-byte and 17 three-byte. With a 16MHz crystal, 58% of the instructions are executed in 0.75μs and 40% in 1.5μs. Multiply and divide instructions require 3μs.



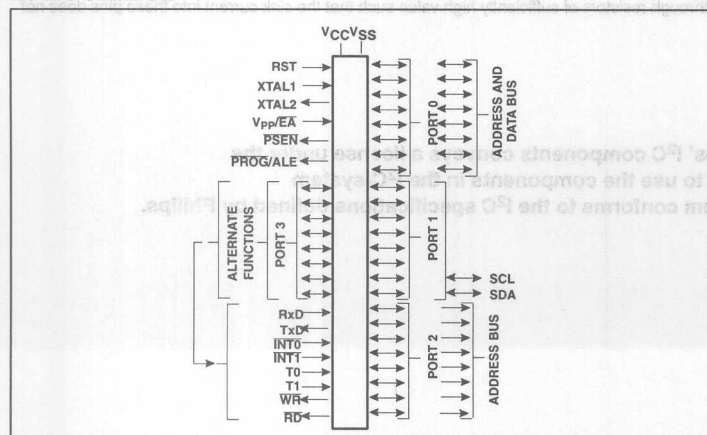
## FEATURES

- 80C51 central processing unit
- 8k × 8 EPROM expandable externally to 64k bytes (EPROM is not expandable)
- 256 × 8 RAM, expandable externally to 64k bytes
- Two standard 16-bit timer/counters
- Four 8-bit I/O ports
- I<sup>2</sup>C-bus serial I/O port with byte oriented master and slave functions
- Full-duplex UART facilities
- Power control modes
  - Idle mode
  - Power-down mode
- Five package styles
- Extended temperature range
- OTP package available
- Three speed ranges
  - 16MHz
  - 20MHz

## PIN CONFIGURATION



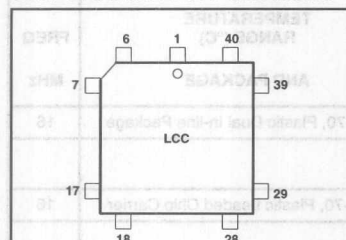
## LOGIC SYMBOL





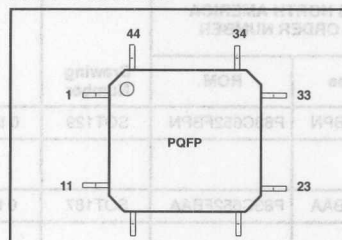
## CMOS single-chip 8-bit microcontroller 87C652

## CERAMIC AND PLASTIC LEADED CHIP CARRIER PIN FUNCTIONS

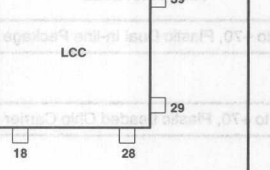
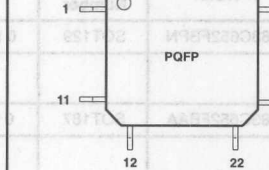


Pin	Function	Pin	Function
1	NC	23	NC
2	P1.0	24	P2.0/A8
3	P1.1	25	P2.1/A9
4	P1.2	26	P2.2/A10
5	P1.3	27	P2.3/A11
6	P1.4	28	P2.4/A12
7	P1.5	29	P2.5/A13
8	P1.6/SCL	30	P2.6/A14
9	P1.7/SDA	31	P2.7/A15
10	RST	32	PSEN
11	P3.0/RxD	33	ALE/PROG
12	NC	34	NC
13	P3.1/TxD	35	EA/Vpp
14	P3.2/INT0	36	P0.7/AD7
15	P3.3/INT1	37	P0.6/AD6
16	P3.4/I/O	38	P0.5/AD5
17	P3.5/I/O	39	P0.4/AD4
18	P3.6/WR	40	P0.3/AD3
19	P3.7/RD	41	P0.2/AD2
20	XTAL2	42	P0.1/AD1
21	XTAL1	43	P0.0/AD0
22	Vcc	44	Vcc

## PLASTIC QUAD FLAT PACK PIN FUNCTIONS



Pin	Function	Pin	Function
1	P1.5	23	P2.5/A13
2	P1.6/SCL	24	P2.6/A14
3	P1.7/SDA	25	P2.7/A15
4	RST	26	PSEN
5	P3.0/RxD	27	ALE/PROG
6	NC	28	NC
7	P3.1/TxD	29	EA/Vpp
8	P3.2/INT0	30	P0.7/AD7
9	P3.3/INT1	31	P0.6/AD6
10	P3.4/T0	32	P0.5/AD5
11	P3.5/T1	33	P0.4/AD4
12	P3.6/WR	34	P0.3/AD3
13	P3.7/RD	35	P0.2/AD2
14	XTAL2	36	P0.1/AD1
15	XTAL1	37	P0.0/AD0
16	VSS	38	VCC
17	NC	39	NC
18	P2.0/A8	40	P1.0
19	P2.1/A9	41	P1.1
20	P2.2/A10	42	P1.2
21	P2.3/A11	43	P1.3
22	P2.4/A12	44	P1.4

CHIP CARRIER PIN FUNCTIONS				PIN FUNCTIONS			
							
Pin	Function	Pin	Function	Pin	Function	Pin	Function
1	NC	23	NC	1	P1.5	23	P2.5/A13
2	P1.0	24	P2.0/A8	2	P1.6/SCL	24	P2.6/A14
3	P1.1	25	P2.1/A9	3	P1.7/SDA	25	P2.7/A15
4	P1.2	26	P2.2/A10	4	RST	26	PSEN
5	P1.3	27	P2.3/A11	5	P3.0/RxD	27	ALE/PROG
6	P1.4	28	P2.4/A12	6	NC	28	NC
7	P1.5	29	P2.5/A13	7	P3.1/TxD	29	EA/Vpp
8	P1.6/SCL	30	P2.6/A14	8	P3.2/INT0	30	P0.7/AD7
9	P1.7/SDA	31	P2.7/A15	9	P3.3/INTT	31	P0.6/AD6
10	RST	32	PSEN	10	P3.4/T0	32	P0.5/AD5
11	P3.0/RxD	33	ALE/PROG	11	P3.5/T1	33	P0.4/AD4
12	NC	34	NC	12	P3.6/WR	34	P0.3/AD3
13	P3.1/TxD	35	EA/Vpp	13	P3.7/RD	35	P0.2/AD2
14	P3.2/INT0	36	P0.7/AD7	14	XTAL2	36	P0.1/AD1
15	P3.3/INTT	37	P0.6/AD6	15	XTAL1	37	P0.0/AD0
16	P3.4/T0	38	P0.5/AD5	16	VSS	38	VCC
17	P3.5/T1	39	P0.4/AD4	17	NC	39	NC
18	P3.6/WR	40	P0.3/AD3	18	P2.0/A8	40	P1.0
19	P3.7/RD	41	P0.2/AD2	19	P2.1/A9	41	P1.1
20	XTAL2	42	P0.1/AD1	20	P2.2/A10	42	P1.2
21	XTAL1	43	P0.0/AD0	21	P2.3/A11	43	P1.3
22	VSS	44	VCC	22	P2.4/A12	44	P1.4

## CMOS single-chip 8-bit microcontroller 87C652

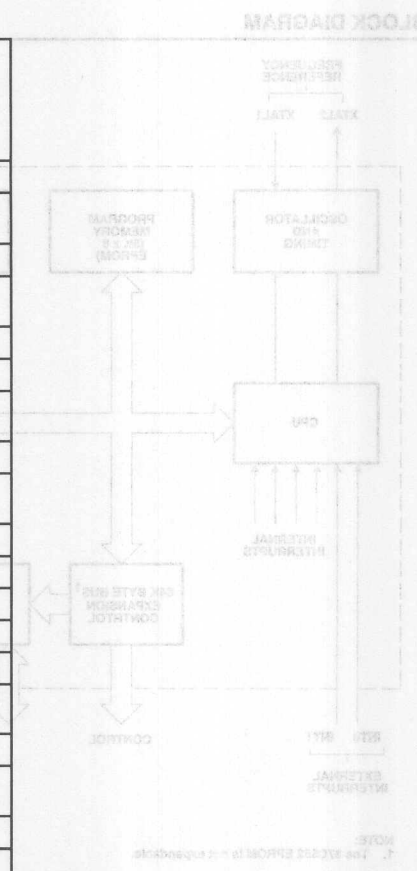
## ORDER INFORMATION

PHILIPS PART ORDER NUMBER PART MARKING		PHILIPS NORTH AMERICA PART ORDER NUMBER		Drawing Number	TEMPERATURE RANGE (°C)  AND PACKAGE	FREQ  MHz
ROMless	ROM	ROMless	ROM			
P80C652FBP	P83C652FBP/xxx	P80C652FBPN	P83C652FBPN	SOT129	0 to +70, Plastic Dual In-line Package	16
P80C652FBA	P83C652FBA/xxx	P80C652FBAA	P83C652FBAA	SOT187	0 to +70, Plastic Leaded Chip Carrier	16
P80C652FBB	P83C652FBB/xxx	P80C652FBBB	P83C652FBBB	SOT307-2 <sup>5</sup>	0 to +70, Plastic Quad Flat Pack	16
P80C652FFP	P83C652FFP/xxx	P80C652FFPN	P83C652FFPN	SOT129	-40 to +85, Plastic Dual In-line Package	16
P80C652FFA	P83C652FFA/xxx	P80C652FFAA	P83C652FFAA	SOT187	-40 to +85, Plastic Leaded Chip Carrier	16
P80C652FFB	P83C652FFB/xxx	P80C652FFBB	P83C652FFBB	SOT307-2 <sup>5</sup>	-40 to +85, Plastic Quad Flat Pack	16
P80C652FHP	P83C652FHP/xxx	P80C652FHPN	P83C652FHPN	SOT129	-40 to +125, Plastic Dual In-line Package	16
P80C652FHA	P83C652FHA/xxx	P80C652FHAA	P83C652FHAA	SOT187	-40 to +125, Plastic Leaded Chip Carrier	16
P80C652FHB	P83C652FHB/xxx	P80C652FHBB	P83C652FHBB	SOT307-2 <sup>5</sup>	-40 to +125, Plastic Quad Flat Pack	16
P80C652IBP	P83C652IBP/xxx	P80C652IBPN	P83C652IBPN	SOT129	0 to +70, Plastic Dual In-line Package	24
P80C652IBA	P83C652IBA/xxx	P80C652IBAA	P83C652IBAA	SOT187	0 to +70, Plastic Leaded Chip Carrier	24
P80C652IBB	P83C652IBB/xxx	P80C652IBBB	P83C652IBBB	SOT307-2 <sup>5</sup>	0 to +70, Plastic Quad Flat Pack	24
P80C652IFP	P83C652IFP/xxx	P80C652IFPN	P83C652IFPN	SOT129	-40 to +85, Plastic Dual In-line Package	24
P80C652IFA	P83C652IFA/xxx	P80C652IFAA	P83C652IFAA	SOT187	-40 to +85, Plastic Leaded Chip Carrier	24
P80C652IFB	P83C652IFB/xxx	P80C652IFBB	P83C652IFBB	SOT307-2 <sup>5</sup>	-40 to +85, Plastic Quad Flat Pack	24

## NOTES:

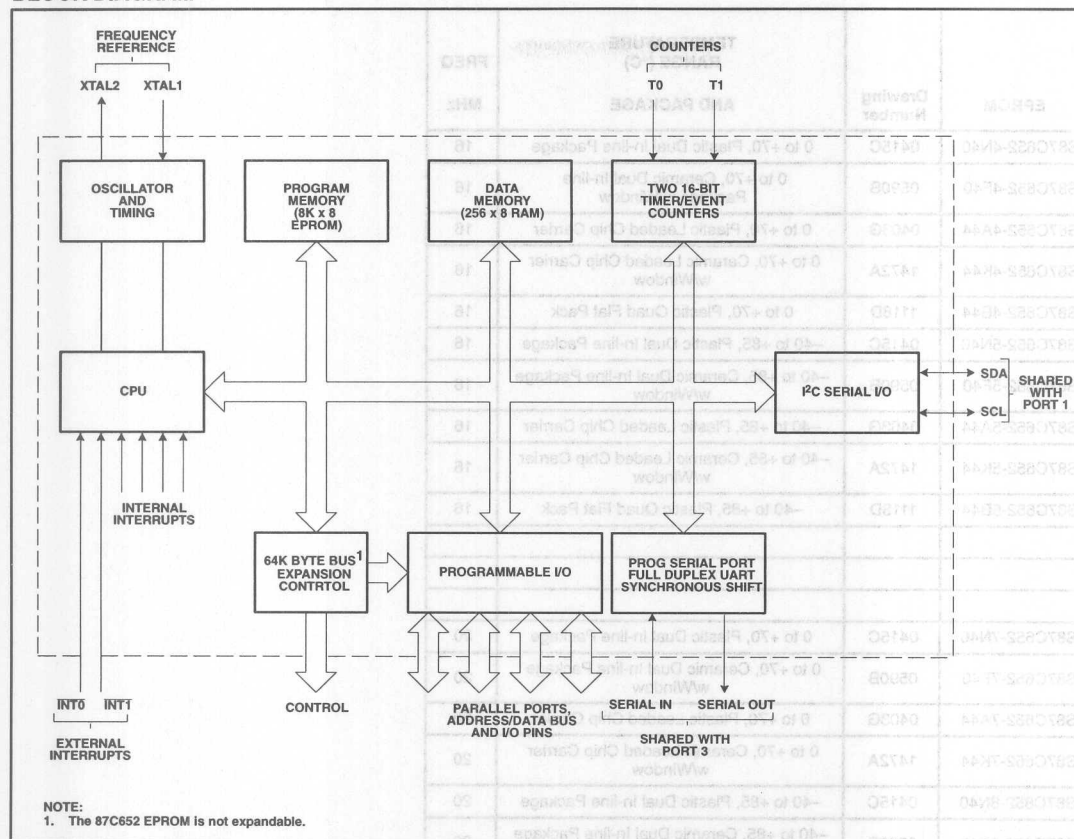
1. 80C652 and 83C652 frequency range is 1.2MHz–16MHz or 1.2 to 24MHz.
2. 87C652 frequency range is 3.5MHz–16MHz or 3.5MHz–20MHz.
3. The 87C652 EPROM is not expandable.
4. xxx denotes the ROM code number.
5. SOT311 replaced by SOT307-2.

## CMOS single-chip 8-bit microcontroller 87C652

[illegible]

## CMOS single-chip 8-bit microcontroller 87C652

## BLOCK DIAGRAM



## CMOS single-chip 8-bit microcontroller 87C652

## PIN DESCRIPTION

MNEMONIC	PIN NO.			TYPE	NAME AND FUNCTION
	DIP	LCC	QFP		
V <sub>SS</sub>	20	22	16	I	<b>Ground:</b> 0V reference.
V <sub>CC</sub>	40	44	38	I	<b>Power Supply:</b> This is the power supply voltage for normal, idle, and power-down operation.
P0.0–0.7	39–32	43–36	37–30	I/O	<b>Port 0:</b> Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s. Port 0 also outputs the code bytes during program verification in the 87C652. External pull-ups are required during program verification.
P1.0–P1.7	1–8	2–9	40–44, 1–3	I/O	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pull-ups, except P1.6 and P1.7 which are open drain. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 1 also receives the low-order address byte during program memory verification. Alternate functions include:
P1.6	7	8	2	I/O	<b>SCL:</b> I <sup>2</sup> C-bus serial port clock line.
P1.7	8	9	3	I/O	<b>SDA:</b> I <sup>2</sup> C-bus serial port data line.
P2.0–P2.7	21–28	24–31	18–25	I/O	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register.
P3.0–P3.7	10–17	11, 13–19	5, 7–13	I/O	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 3 also serves the special features of the 80C51 family, as listed below:
	10	11	5	I	<b>RxD (P3.0):</b> Serial input port
	11	13	7	O	<b>TxD (P3.1):</b> Serial output port
	12	14	8	I	<b>INT0 (P3.2):</b> External interrupt
	13	15	9	I	<b>INT1 (P3.3):</b> External interrupt
	14	16	10	I	<b>T0 (P3.4):</b> Timer 0 external input
	15	17	11	I	<b>T1 (P3.5):</b> Timer 1 external input
	16	18	12	O	<b>WR (P3.6):</b> External data memory write strobe
	17	19	13	O	<b>RD (P3.7):</b> External data memory read strobe
RST	9	10	4	I	<b>Reset:</b> A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to V <sub>SS</sub> permits a power-on reset using only an external capacitor to V <sub>CC</sub> .
ALE/PROG	30	33	27	I/O	<b>Address Latch Enable/Program Pulse:</b> Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory. This pin is also the program pulse input (PROG) during EPROM programming.
PSEN	29	32	26	O	<b>Program Store Enable:</b> The read strobe to external program memory. When the 87C652 is executing code from the external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory. PSEN is not activated during fetches from internal program memory.
EA/V <sub>PP</sub>	31	35	29	I	<b>External Access Enable/Programming Supply Voltage:</b> EA must be externally held low to enable the device to fetch code from external program memory locations 0000H and 1FFFH. If EA is held high, the device executes from internal program memory unless the program counter contains an address greater than 1FFFH. This pin also receives the 12.75V programming supply voltage (V <sub>PP</sub> ) during EPROM programming.
XTAL1	19	21	15	I	<b>Crystal 1:</b> Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	20	14	O	<b>Crystal 2:</b> Output from the inverting oscillator amplifier.

## NOTE:

To avoid "latch-up" effect at power-on, the voltage on any pin at any time must not be higher than V<sub>CC</sub> + 0.5V or V<sub>SS</sub> – 0.5V, respectively.



## CMOS single-chip 8-bit microcontroller 87C652

Table 1. 8XC652/654 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB							LSB	
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
DPTR:	Data pointer (2 bytes)										
DPH	Data pointer high	83H									00H
DPL	Data pointer low	82H									00H
IE*#	Interrupt enable	A8H	AF	AE	AD	AC	AB	AA	A9	A8	0x000000B
			EA		ES1	ES0	ET1	EX1	ET0	EX0	
IP*#	Interrupt priority	B8H	—		PS1	PS0	PT1	PX1	PT0	PX0	xx000000B
			87	86	85	84	83	82	81	80	
P0*	Port 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
			97	96	95	94	93	92	91	90	
P1*#	Port 1	90H	SDA	SCL							FFH
			A7	A6	A5	A4	A3	A2	A1	A0	
P2*	Port 2	A0H	A15	A14	A13	A12	A11	A10	A9	A8	FFH
			B7	B6	B5	B4	B3	B2	B1	B0	
P3*	Port 3	B0H	RD	WR	T1	T0	INT1	INT0	TXD	RXD	FFH
PCON	Power control	87H	SMOD	—	—	—	GF1	GF0	PD	IDL	0xxx0000B
			9F	9E	9D	9C	9B	9A	99	98	
S0CON*#	Serial 0 port control	98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00H
S0BUF#	Serial 0 data buffer	99H									xxxxxxxB
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program status word	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	00H
S1DAT#	Serial 1 data	DAH									00H
SP	Stack pointer	81H									07H
S1ADR#	Serial 1 address	DBH	SLAVE ADDRESS							GC	00H
S1STA#	Serial 1 status	D9H	SC4	SC3	SC2	SC1	SC0	0	0	0	F8H
			DF	DE	DD	DC	DB	DA	D9	D8	
S1CON*#	Serial 1 control	D8H	CR2	ENS1	STA	STO	SI	AA	CR1	CR0	00000000B
			8F	8E	8D	8C	8B	8A	89	88	
TCON*	Timer control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
TH1	Timer high 1	8DH									00H
TH0	Timer high 0	8CH									00H
TL1	Timer low 1	8BH									00H
TH0	Timer low 0	8AH									00H
TMOD	Timer mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.

## CMOS single-chip 8-bit microcontroller 87C652

**OSCILLATOR CHARACTERISTICS**

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator, as shown in the logic symbol, page 934.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

**RESET**

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. To insure a good power-on reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few

milliseconds) plus two machine cycles. At power-on, the voltage on V<sub>CC</sub> and RST must come up at the same time for a proper start-up.

**IDLE MODE**

In the idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

**POWER-DOWN MODE**

In the power-down mode, the oscillator is stopped and the instruction to invoke

power-down is the last instruction executed. Only the contents of the on-chip RAM are preserved. A hardware reset is the only way to terminate the power-down mode: the control bits for the reduced power modes are in the special function register PCON. Table 2 shows the state of the I/O ports during low current operating modes.

**I<sup>2</sup>C SERIAL COMMUNICATION — SIO1**

The I<sup>2</sup>C serial port is identical to the I<sup>2</sup>C serial port on the 8XC552. The operation of this subsystem is described in detail in the 8XC552 section of this manual.

Note that in both the 8XC652/4 and the 8XC552 the I<sup>2</sup>C pins are alternate functions to port pins P1.6 and P1.7. Because of this, P1.6 and P1.7 on these parts do not have a pull-up structure as found on the 80C51. Therefore P1.6 and P1.7 have open drain outputs on the 8XC652/4.

**Table 2. External Pin Status During Idle and Power-Down Mode**

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

**Serial Control Register (S1CON) – See Table 3**

S1CON (D8H) CR2 ENS1 STA STO SI AA CR1 CR0

Bits CR0, CR1 and CR2 determine the serial clock frequency that is generated in the master mode of operation.

**Table 3. Serial Clock Rates**

CR2	CR1	CR0	BIT FREQUENCY (kHz) AT f <sub>osc</sub>				f <sub>osc</sub> DIVIDED BY
			6MHz	12MHz	16MHz	20MHz	
0	0	0	23	47	62.5	78	256
0	0	1	27	54	71	89	224
0	1	0	31.25	62.5	83.3	104 <sup>1</sup>	192
0	1	1	37	75	100	125 <sup>1</sup>	160
1	0	0	6.25	12.5	17	21	960
1	0	1	50	100	133 <sup>1</sup>	166 <sup>1</sup>	120
1	1	0	100	200 <sup>1</sup>	267 <sup>1</sup>	334 <sup>1</sup>	60
1	1	1	> 0.25 < 62.5 0 to 255	> 0.5 < 62.5 0 to 254	> 0.65 < 55.6 0 to 253	> 0.81 < 69.4 0 to 253	96 × (256 – (reload value Timer 1)) (Reload value range: 0 – 254 in mode 2)

**NOTES:**

1. These frequencies exceed the upper limit of 100kHz of the I<sup>2</sup>C-bus specification and cannot be used in an I<sup>2</sup>C-bus application.

## CMOS single-chip 8-bit microcontroller 87C652

ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>

PARAMETER	RATING	UNIT
Storage temperature range	-65 to +150	°C
Voltage on $\overline{EA}/V_{PP}$ to $V_{SS}$	-0.5 to +13	V
Voltage on any other pin to $V_{SS}$	-0.5 to +6.5	V
Input, output current on any single pin	$\pm 5$	mA
Input, output current on any two pins	$\pm 10$	mA
Power dissipation (based on package heat transfer limitations, not device power consumption)	1	W

## NOTES:

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
- This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
- Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to  $V_{SS}$  unless otherwise noted.

## DEVICE SPECIFICATIONS

TYPE	SUPPLY VOLTAGE (V)		FREQUENCY (MHz)		TEMPERATURE (°C)
	MIN.	MAX.	MIN.	MAX.	
S87C652-4	4.5	5.5	3.5	16	0 to +70
S87C652-5	4.5	5.5	3.5	16	-40 to +85
S87C652-7	4.5	5.5	3.5	20	0 to +70
S87C652-8	4.5	5.5	3.5	20	-40 to +85

Serial Control Register (SCON) - See Table 3

CR0	CR1	CR2	CR3	CR4	CR5	CR6	CR7	CR8	CR9	CR10	CR11	CR12	CR13	CR14	CR15	CR16	CR17	CR18	CR19	CR20	CR21	CR22	CR23	CR24	CR25	CR26	CR27	CR28	CR29	CR30	CR31	CR32	CR33	CR34	CR35	CR36	CR37	CR38	CR39	CR40	CR41	CR42	CR43	CR44	CR45	CR46	CR47	CR48	CR49	CR50	CR51	CR52	CR53	CR54	CR55	CR56	CR57	CR58	CR59	CR60	CR61	CR62	CR63	CR64	CR65	CR66	CR67	CR68	CR69	CR70	CR71	CR72	CR73	CR74	CR75	CR76	CR77	CR78	CR79	CR80	CR81	CR82	CR83	CR84	CR85	CR86	CR87	CR88	CR89	CR90	CR91	CR92	CR93	CR94	CR95	CR96	CR97	CR98	CR99	CR100	CR101	CR102	CR103	CR104	CR105	CR106	CR107	CR108	CR109	CR110	CR111	CR112	CR113	CR114	CR115	CR116	CR117	CR118	CR119	CR120	CR121	CR122	CR123	CR124	CR125	CR126	CR127	CR128	CR129	CR130	CR131	CR132	CR133	CR134	CR135	CR136	CR137	CR138	CR139	CR140	CR141	CR142	CR143	CR144	CR145	CR146	CR147	CR148	CR149	CR150	CR151	CR152	CR153	CR154	CR155	CR156	CR157	CR158	CR159	CR160	CR161	CR162	CR163	CR164	CR165	CR166	CR167	CR168	CR169	CR170	CR171	CR172	CR173	CR174	CR175	CR176	CR177	CR178	CR179	CR180	CR181	CR182	CR183	CR184	CR185	CR186	CR187	CR188	CR189	CR190	CR191	CR192	CR193	CR194	CR195	CR196	CR197	CR198	CR199	CR200	CR201	CR202	CR203	CR204	CR205	CR206	CR207	CR208	CR209	CR210	CR211	CR212	CR213	CR214	CR215	CR216	CR217	CR218	CR219	CR220	CR221	CR222	CR223	CR224	CR225	CR226	CR227	CR228	CR229	CR230	CR231	CR232	CR233	CR234	CR235	CR236	CR237	CR238	CR239	CR240	CR241	CR242	CR243	CR244	CR245	CR246	CR247	CR248	CR249	CR250	CR251	CR252	CR253	CR254	CR255	CR256	CR257	CR258	CR259	CR260	CR261	CR262	CR263	CR264	CR265	CR266	CR267	CR268	CR269	CR270	CR271	CR272	CR273	CR274	CR275	CR276	CR277	CR278	CR279	CR280	CR281	CR282	CR283	CR284	CR285	CR286	CR287	CR288	CR289	CR290	CR291	CR292	CR293	CR294	CR295	CR296	CR297	CR298	CR299	CR300	CR301	CR302	CR303	CR304	CR305	CR306	CR307	CR308	CR309	CR310	CR311	CR312	CR313	CR314	CR315	CR316	CR317	CR318	CR319	CR320	CR321	CR322	CR323	CR324	CR325	CR326	CR327	CR328	CR329	CR330	CR331	CR332	CR333	CR334	CR335	CR336	CR337	CR338	CR339	CR340	CR341	CR342	CR343	CR344	CR345	CR346	CR347	CR348	CR349	CR350	CR351	CR352	CR353	CR354	CR355	CR356	CR357	CR358	CR359	CR360	CR361	CR362	CR363	CR364	CR365	CR366	CR367	CR368	CR369	CR370	CR371	CR372	CR373	CR374	CR375	CR376	CR377	CR378	CR379	CR380	CR381	CR382	CR383	CR384	CR385	CR386	CR387	CR388	CR389	CR390	CR391	CR392	CR393	CR394	CR395	CR396	CR397	CR398	CR399	CR400	CR401	CR402	CR403	CR404	CR405	CR406	CR407	CR408	CR409	CR410	CR411	CR412	CR413	CR414	CR415	CR416	CR417	CR418	CR419	CR420	CR421	CR422	CR423	CR424	CR425	CR426	CR427	CR428	CR429	CR430	CR431	CR432	CR433	CR434	CR435	CR436	CR437	CR438	CR439	CR440	CR441	CR442	CR443	CR444	CR445	CR446	CR447	CR448	CR449	CR450	CR451	CR452	CR453	CR454	CR455	CR456	CR457	CR458	CR459	CR460	CR461	CR462	CR463	CR464	CR465	CR466	CR467	CR468	CR469	CR470	CR471	CR472	CR473	CR474	CR475	CR476	CR477	CR478	CR479	CR480	CR481	CR482	CR483	CR484	CR485	CR486	CR487	CR488	CR489	CR490	CR491	CR492	CR493	CR494	CR495	CR496	CR497	CR498	CR499	CR500	CR501	CR502	CR503	CR504	CR505	CR506	CR507	CR508	CR509	CR510	CR511	CR512	CR513	CR514	CR515	CR516	CR517	CR518	CR519	CR520	CR521	CR522	CR523	CR524	CR525	CR526	CR527	CR528	CR529	CR530	CR531	CR532	CR533	CR534	CR535	CR536	CR537	CR538	CR539	CR540	CR541	CR542	CR543	CR544	CR545	CR546	CR547	CR548	CR549	CR550	CR551	CR552	CR553	CR554	CR555	CR556	CR557	CR558	CR559	CR560	CR561	CR562	CR563	CR564	CR565	CR566	CR567	CR568	CR569	CR570	CR571	CR572	CR573	CR574	CR575	CR576	CR577	CR578	CR579	CR580	CR581	CR582	CR583	CR584	CR585	CR586	CR587	CR588	CR589	CR590	CR591	CR592	CR593	CR594	CR595	CR596	CR597	CR598	CR599	CR600	CR601	CR602	CR603	CR604	CR605	CR606	CR607	CR608	CR609	CR610	CR611	CR612	CR613	CR614	CR615	CR616	CR617	CR618	CR619	CR620	CR621	CR622	CR623	CR624	CR625	CR626	CR627	CR628	CR629	CR630	CR631	CR632	CR633	CR634	CR635	CR636	CR637	CR638	CR639	CR640	CR641	CR642	CR643	CR644	CR645	CR646	CR647	CR648	CR649	CR650	CR651	CR652	CR653	CR654	CR655	CR656	CR657	CR658	CR659	CR660	CR661	CR662	CR663	CR664	CR665	CR666	CR667	CR668	CR669	CR670	CR671	CR672	CR673	CR674	CR675	CR676	CR677	CR678	CR679	CR680	CR681	CR682	CR683	CR684	CR685	CR686	CR687	CR688	CR689	CR690	CR691	CR692	CR693	CR694	CR695	CR696	CR697	CR698	CR699	CR700	CR701	CR702	CR703	CR704	CR705	CR706	CR707	CR708	CR709	CR710	CR711	CR712	CR713	CR714	CR715	CR716	CR717	CR718	CR719	CR720	CR721	CR722	CR723	CR724	CR725	CR726	CR727	CR728	CR729	CR730	CR731	CR732	CR733	CR734	CR735	CR736	CR737	CR738	CR739	CR740	CR741	CR742	CR743	CR744	CR745	CR746	CR747	CR748	CR749	CR750	CR751	CR752	CR753	CR754	CR755	CR756	CR757	CR758	CR759	CR760	CR761	CR762	CR763	CR764	CR765	CR766	CR767	CR768	CR769	CR770	CR771	CR772	CR773	CR774	CR775	CR776	CR777	CR778	CR779	CR780	CR781	CR782	CR783	CR784	CR785	CR786	CR787	CR788	CR789	CR790	CR791	CR792	CR793	CR794	CR795	CR796	CR797	CR798	CR799	CR800	CR801	CR802	CR803	CR804	CR805	CR806	CR807	CR808	CR809	CR810	CR811	CR812	CR813	CR814	CR815	CR816	CR817	CR818	CR819	CR820	CR821	CR822	CR823	CR824	CR825	CR826	CR827	CR828	CR829	CR830	CR831	CR832	CR833	CR834	CR835	CR836	CR837	CR838	CR839	CR840	CR841	CR842	CR843	CR844	CR845	CR846	CR847	CR848	CR849	CR850	CR851	CR852	CR853	CR854	CR855	CR856	CR857	CR858	CR859	CR860	CR861	CR862	CR863	CR864	CR865	CR866	CR867	CR868	CR869	CR870	CR871	CR872	CR873	CR874	CR875	CR876	CR877	CR878	CR879	CR880	CR881	CR882	CR883	CR884	CR885	CR886	CR887	CR888	CR889	CR890	CR891	CR892	CR893	CR894	CR895	CR896	CR897	CR898	CR899	CR900	CR901	CR902	CR903	CR904	CR905	CR906	CR907	CR908	CR909	CR910	CR911	CR912	CR913	CR914	CR915	CR916	CR917	CR918	CR919	CR920	CR921	CR922	CR923	CR924	CR925	CR926	CR927	CR928	CR929	CR930	CR931	CR932	CR933	CR934	CR935	CR936	CR937	CR938	CR939	CR940	CR941	CR942	CR943	CR944	CR945	CR946	CR947	CR948	CR949	CR950	CR951	CR952	CR953	CR954	CR955	CR956	CR957	CR958	CR959	CR960	CR961	CR962	CR963	CR964	CR965	CR966	CR967	CR968	CR969	CR970	CR971	CR972	CR973	CR974	CR975	CR976	CR977	CR978	CR979	CR980	CR981	CR982	CR983	CR984	CR985	CR986	CR987	CR988	CR989	CR990	CR991	CR992	CR993	CR994	CR995	CR996	CR997	CR998	CR999	CR1000	CR1001	CR1002	CR1003	CR1004	CR1005	CR1006	CR1007	CR1008	CR1009	CR1010	CR1011	CR1012	CR1013	CR1014	CR1015	CR1016	CR1017	CR1018	CR1019	CR1020	CR1021	CR1022	CR1023	CR1024	CR1025	CR1026	CR1027	CR1028	CR1029	CR1030	CR1031	CR1032	CR1033	CR1034	CR1035	CR1036	CR1037	CR1038	CR1039	CR1040	CR1041	CR1042	CR1043	CR1044	CR1045	CR1046	CR1047	CR1048	CR1049	CR1050	CR1051	CR1052	CR1053	CR1054	CR1055	CR1056	CR1057	CR1058	CR1059	CR1060	CR1061	CR1062	CR1063	CR1064	CR1065	CR1066	CR1067	CR1068	CR1069	CR1070	CR1071	CR1072	CR1073	CR1074	CR1075	CR1076	CR1077	CR1078	CR1079	CR1080	CR1081	CR1082	CR1083	CR1084	CR1085	CR1086	CR1087	CR1088	CR1089	CR1090	CR1091	CR1092	CR1093	CR1094	CR1095	CR1096	CR1097	CR1098	CR1099	CR1100	CR1101	CR1102	CR1103	CR1104	CR1105	CR1106	CR1107	CR1108	CR1109	CR1110	CR1111	CR1112	CR1113	CR1114	CR1115	CR1116	CR1117	CR1118	CR1119	CR1120	CR1121	CR1122	CR1123	CR1124	CR1125	CR1126	CR1127	CR1128	CR1129	CR1130	CR1131	CR1132	CR1133	CR1134	CR1135	CR1136	CR1137	CR1138	CR1139	CR1140	CR1141	CR1142	CR1143	CR1144	CR1145	CR1146	CR1147	CR1148	CR1149	CR1150	CR1151	CR1152	CR1153	CR1154	CR1155	CR1156	CR1157	CR1158	CR1159	CR1160	CR1161	CR1162	CR1163	CR1164	CR1165	CR1166	CR1167	CR1168	CR1169	CR1170	CR1171	CR1172	CR1173	CR1174	CR1175	CR1176	CR1177	CR1178	CR1179	CR1180	CR1181	CR1182	CR1183	CR1184	CR1185	CR1186	CR1187	CR1188	CR1189	CR1190	CR1191	CR1192	CR1193	CR1194	CR1195	CR1196	CR1197	CR1198	CR1199	CR1200	CR1201	CR1202	CR1203	CR1204	CR1205	CR1206	CR1207	CR1208	CR1209	CR1210	CR1211	CR1212	CR1213	CR1214	CR1215	CR1216	CR1217	CR1218	CR1219	CR1220	CR1221	CR1222	CR1223	CR1224	CR1225	CR1226	CR1227	CR1228	CR1229	CR1230	CR1231	CR1232	CR1233	CR1234	CR1235	CR1236	CR1237	CR1238	CR1239	CR1240	CR1241	CR1242	CR1243	CR1244	CR1245	CR1246	CR1247	CR1248	CR1249	CR1250	CR1251	CR1252	CR1253	CR1254	CR1255	CR1256	CR1257	CR1258	CR1259	CR1260	CR1261	CR1262	CR1263	CR1264	CR1265	CR1266	CR1267	CR1268	CR1269	CR1270	CR1271	CR1272	CR1273	CR1274	CR1275	CR1276	CR1277	CR1278	CR1279	CR1280	CR1281	CR1282	CR1283	CR1284	CR1285	CR1286	CR1287	CR1288	CR1289	CR1290	CR1291	CR1292	CR1293	CR1294	CR1295	CR1296	CR1297	CR1298	CR1299	CR1300	CR1301	CR1302	CR1303	CR1304	CR1305	CR1306	CR1307	CR1308	CR1309	CR1310	CR1311	CR1312	CR1313	CR1314	CR1315	CR1316	CR1317	CR1318	CR1319	CR1320	CR1321	CR1322	CR1323	CR1324	CR1325	CR1326	CR1327	CR1328	CR1329	CR1330	CR1331	CR1332	CR1333	CR1334	CR1335	CR1336	CR1337	CR1338	CR1339	CR1340	CR1341	CR1342	CR1343	CR1344	CR1345	CR1346	CR1347	CR1348
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------

## CMOS single-chip 8-bit microcontroller 87C652

## DC ELECTRICAL CHARACTERISTICS

 $V_{SS} = 0V$ 

SYMBOL	PARAMETER	PART TYPE	TEST CONDITIONS	LIMITS		UNIT
				MIN.	MAX.	
$V_{IL}$	Input low voltage, except $\overline{EA}$ , P1.6/SCL, P1.7/SDA	0°C to +70°C -40°C to +85°C		-0.5 -0.5	$0.2V_{CC} - 0.1$ $0.2V_{CC} - 0.15$	V V
$V_{IL1}$	Input low voltage to $\overline{EA}$	0°C to +70°C -40°C to +85°C		-0.5 -0.5	$0.2V_{CC} - 0.3$ $0.2V_{CC} - 0.35$	V V
$V_{IL2}$	Input low voltage to P1.6/SCL, P1.7/SDA <sup>6</sup>			-0.5	1.5	V
$V_{IH}$	Input high voltage, except XTAL1, RST, P1.6/SCL, P1.7/SDA	0°C to +70°C -40°C to +85°C		$0.2V_{CC} + 0.9$ $0.2V_{CC} + 1.0$	$V_{CC} + 0.5$ $V_{CC} + 0.5$	V V
$V_{IH1}$	Input high voltage, XTAL1, RST	0°C to +70°C -40°C to +85°C		$0.7V_{CC}$ $0.7V_{CC} + 0.1$	$V_{CC} + 0.5$ $V_{CC} + 0.5$	V V
$V_{IH2}$	Input high voltage, P1.6/SCL, P1.7/SDA <sup>6</sup>			3.0	6.0	V
$V_{OL}$	Output low voltage, ports 1, 2, 3, except P1.6/SCL, P1.7/SDA		$I_{OL} = 1.6mA^8$		0.45	V
$V_{OL1}$	Output low voltage, port 0, ALE, $\overline{PSEN}$		$I_{OL} = 3.2mA^8$		0.45	V
$V_{OL2}$	Output low voltage, P1.6/SCL, P1.7/SDA		$I_{OL} = 3.0mA$		0.4	V
$V_{OH}$	Output high voltage, ports 1, 2, 3	0°C to +70°C -40°C to +85°C	$I_{OH} = -60\mu A$ $I_{OH} = -25\mu A$	2.4 $0.75V_{CC}$		V V
$V_{OH1}$	Output high voltage, Port 0 in external bus mode, ALE, $\overline{PSEN}$ , RST <sup>9</sup>	0°C to +70°C -40°C to +85°C	$I_{OH} = -400\mu A$ $I_{OH} = -150\mu A$	2.4 $0.75V_{CC}$		V V
$I_{IL}$	Logical 0 input current, ports 1, 2, 3, except P1.6/SCL, P1.7/SDA	0°C to +70°C -40°C to +85°C	$V_{IN} = 0.45V$		-50 -75	$\mu A$ $\mu A$
$I_{TL}$	Logical 1-to-0 transition current, ports 1, 2, 3, except P1.6/SCL, P1.7/SDA	0°C to +70°C -40°C to +85°C	See Note 7		-650 -750	$\mu A$ $\mu A$
$I_{L1}$	Input leakage current, port 0		$0.45 < V_i < V_{CC}$		$\pm 10$	$\mu A$
$I_{L2}$	Input leakage current, P1.6/SCL, P1.7/SDA		$0V < V_i < 6.0V$ $0V < V_{CC} < 6.0V$		$\pm 10$	$\mu A$ $\mu A$
$I_{CC}$	Power supply current:		See Note 1 $V_{CC} = 6.0V$			
	Active mode @ 16MHz <sup>2</sup>				25	mA
	Idle mode @ 16MHz <sup>3</sup>				6	mA
	Power down mode <sup>4, 5</sup>	0°C to +70°C			50	$\mu A$
	Power down mode <sup>4, 5</sup>	-40°C to +125°C			135	$\mu A$
$R_{RST}$	Internal reset pull-down resistor			50	150	k $\Omega$
$C_{IO}$	Pin Capacitance		Freq. = 1MHz		10	pF

## NOTES:

- See Figures 9 through 12 for  $I_{CC}$  test conditions.
- The operating supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 10ns$ ;  $V_{IL} = V_{SS} + 0.5V$ ;  $V_{IH} = V_{CC} - 0.5V$ ; XTAL2 not connected;  $\overline{EA} = RST = Port 0 = P1.6 = P1.7 = V_{CC}$ ;  $f_{CLK} = 16MHz$ . See Figure 9.
- The idle mode supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 10ns$ ;  $V_{IL} = V_{SS} + 0.5V$ ;  $V_{IH} = V_{CC} - 0.5V$ ; XTAL2 not connected; Port 0 = P1.6 = P1.7 =  $V_{CC}$ ;  $\overline{EA} = RST = V_{SS}$ ;  $f_{CLK} = 16MHz$ . See Figure 10.
- The power-down current is measured with all output pins disconnected; XTAL2 not connected; Port 0 = P1.6 = P1.7 =  $V_{CC}$ ;  $\overline{EA} = RST = V_{SS}$ . See Figure 12.
- $2V \leq V_{PD} \leq V_{CCmax}$ .
- The input threshold voltage of P1.6 and P1.7 (SIO1) meets the I<sup>2</sup>C specification, so an input voltage below 1.5V will be recognized as a logic 0 while an input voltage above 3.0V will be recognized as a logic 1.
- Pins of ports 1, 2, and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when  $V_{IN}$  is approximately 2V.
- Capacitive loading on ports 0 and 2 may cause spurious noise to be superimposed on the  $V_{OLs}$  of ALE and ports 1 and 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100pF), the noise pulse on the ALE pin may exceed 0.8V. In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.  $I_{OL}$  can exceed these conditions provided that no single output sinks more than 5mA and no more than two outputs exceed the test conditions.
- Capacitive loading on ports 0 and 2 may cause the  $V_{OH}$  on ALE and  $\overline{PSEN}$  to momentarily fall below the  $0.9V_{CC}$  specification when the address bits are stabilizing.

## CMOS single-chip 8-bit microcontroller 87C652

## AC ELECTRICAL CHARACTERISTICS

T<sub>amb</sub> = 0°C to +70°C, or T<sub>amb</sub> = -40°C to +85°C, V<sub>SS</sub> = 0V<sup>1,2</sup>

SYMBOL	FIGURE	PARAMETER	16MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
1/t <sub>CLCL</sub>	1	Oscillator frequency			3.5	16	MHz
t <sub>LHLL</sub>	1	ALE pulse width	85		2t <sub>CLCL</sub> -40		ns
t <sub>AVLL</sub>	1	Address valid to ALE low	-8		t <sub>CLCL</sub> -55		ns
t <sub>LLAX</sub>	1	Address hold after ALE low	28		t <sub>CLCL</sub> -35		ns
t <sub>LLIV</sub>	1	ALE low to valid instruction in		150		4t <sub>CLCL</sub> -100	ns
t <sub>LLPL</sub>	1	ALE low to PSEN low	23		t <sub>CLCL</sub> -40		ns
t <sub>PLPH</sub>	1	PSEN pulse width	143		3t <sub>CLCL</sub> -45		ns
t <sub>PLIV</sub>	1	PSEN low to valid instruction in		83		3t <sub>CLCL</sub> -105	ns
t <sub>PIX</sub>	1	Input instruction hold after PSEN	0		0		ns
t <sub>PIXZ</sub>	1	Input instruction float after PSEN		38		t <sub>CLCL</sub> -25	ns
t <sub>AVIV</sub>	1	Address to valid instruction in		208		5t <sub>CLCL</sub> -105	ns
t <sub>PLAZ</sub>	1	PSEN low to address float		10		10	ns
<b>Data Memory</b>							
t <sub>AVLL</sub>	2, 3	Address valid to ALE low	28		t <sub>CLCL</sub> -35		ns
t <sub>RLRH</sub>	2, 3	RD pulse width	275		6t <sub>CLCL</sub> -100		ns
t <sub>WLWH</sub>	2, 3	WR pulse width	275		6t <sub>CLCL</sub> -100		ns
t <sub>RLDV</sub>	2, 3	RD low to valid data in		148		5t <sub>CLCL</sub> -165	ns
t <sub>RHDX</sub>	2, 3	Data hold after RD	0		0		ns
t <sub>RHDZ</sub>	2, 3	Data float after RD		55		2t <sub>CLCL</sub> -70	ns
t <sub>LLDV</sub>	2, 3	ALE low to valid data in		350		8t <sub>CLCL</sub> -150	ns
t <sub>AVDV</sub>	2, 3	Address to valid data in		398		9t <sub>CLCL</sub> -165	ns
t <sub>LLWL</sub>	2, 3	ALE low to RD or WR low	138	238	3t <sub>CLCL</sub> -50	3t <sub>CLCL</sub> +50	ns
t <sub>AVWL</sub>	2, 3	Address valid to WR low or RD low	120		4t <sub>CLCL</sub> -130		ns
t <sub>QVWX</sub>	2, 3	Data valid to WR transition	3		t <sub>CLCL</sub> -60		ns
t <sub>DW</sub>	2, 3	Data setup time before WR	288		7t <sub>CLCL</sub> -150		ns
t <sub>WHQX</sub>	2, 3	Data hold after WR	13		t <sub>CLCL</sub> -50		ns
t <sub>RLAZ</sub>	2, 3	RD low to address float		0		0	ns
t <sub>WHLH</sub>	2, 3	RD or WR high to ALE high	23	103	t <sub>CLCL</sub> -40	t <sub>CLCL</sub> +40	ns
<b>Shift Register</b>							
t <sub>XLXL</sub>	4	Serial port clock cycle time <sup>4</sup>	0.75		12t <sub>CLCL</sub>		μs
t <sub>QVXH</sub>	4	Output data setup to clock rising edge <sup>4</sup>	492		10t <sub>CLCL</sub> -133		ns
t <sub>XHQX</sub>	4	Output data hold after clock rising edge <sup>4</sup>	8		2t <sub>CLCL</sub> -117		ns
t <sub>XHDX</sub>	4	Input data hold after clock rising edge <sup>4</sup>	0		0		ns
t <sub>XHDV</sub>	4	Clock rising edge to input data valid <sup>4</sup>		498		10t <sub>CLCL</sub> -133	ns
<b>External Clock</b>							
t <sub>CHCX</sub>	5	High time <sup>4</sup>	20		20	t <sub>CLCL</sub> - t <sub>LOW</sub>	ns
t <sub>CLCX</sub>	5	Low time <sup>4</sup>	20		20	t <sub>CLCL</sub> - t <sub>HIGH</sub>	ns
t <sub>CLCH</sub>	5	Rise time <sup>4</sup>		20		20	ns
t <sub>CHCL</sub>	5	Fall time <sup>4</sup>		20		20	ns



## CMOS single-chip 8-bit microcontroller 87C652

## AC ELECTRICAL CHARACTERISTICS (Continued)

T<sub>amb</sub> = 0°C to +70°C, or T<sub>amb</sub> = -40°C to +85°C, V<sub>SS</sub> = 0V<sup>1,2</sup>

SYMBOL	FIGURE	PARAMETER	20MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
1/t <sub>CLCL</sub>	1	Oscillator frequency			3.5	20	MHz
t <sub>LHLL</sub>	1	ALE pulse width	60		2t <sub>CLCL</sub> -40		ns
t <sub>AVLL</sub>	1	Address valid to ALE low	25		t <sub>CLCL</sub> -25		ns
t <sub>LLAX</sub>	1	Address hold after ALE low	25		t <sub>CLCL</sub> -25		ns
t <sub>LLIV</sub>	1	ALE low to valid instruction in		135		4t <sub>CLCL</sub> -65	ns
t <sub>LLPL</sub>	1	ALE low to PSEN low	25		t <sub>CLCL</sub> -25		ns
t <sub>PLPH</sub>	1	PSEN pulse width	105		3t <sub>CLCL</sub> -45		ns
t <sub>PLIV</sub>	1	PSEN low to valid instruction in		90		3t <sub>CLCL</sub> -60	ns
t <sub>PIXI</sub>	1	Input instruction hold after PSEN	0		0		ns
t <sub>PIXZ</sub>	1	Input instruction float after PSEN		25		t <sub>CLCL</sub> -25	ns
t <sub>AVIV</sub>	1	Address to valid instruction in		170		5t <sub>CLCL</sub> -80	ns
t <sub>PLAZ</sub>	1	PSEN low to address float		10		10	ns
<b>Data Memory</b>							
t <sub>AVLL</sub>	2, 3	Address valid to ALE low	25		t <sub>CLCL</sub> -25		ns
t <sub>RLRH</sub>	2, 3	RD pulse width	200		6t <sub>CLCL</sub> -100		ns
t <sub>WLWH</sub>	2, 3	WR pulse width	200		6t <sub>CLCL</sub> -100		ns
t <sub>RLDV</sub>	2, 3	RD low to valid data in		160		5t <sub>CLCL</sub> -90	ns
t <sub>RHDX</sub>	2, 3	Data hold after RD	0		0		ns
t <sub>RHDZ</sub>	2, 3	Data float after RD		72		2t <sub>CLCL</sub> -28	ns
t <sub>LLDV</sub>	2, 3	ALE low to valid data in		250		8t <sub>CLCL</sub> -150	ns
t <sub>AVDV</sub>	2, 3	Address to valid data in		285		9t <sub>CLCL</sub> -165	ns
t <sub>LLWL</sub>	2, 3	ALE low to RD or WR low	100	200	3t <sub>CLCL</sub> -50	3t <sub>CLCL</sub> +50	ns
t <sub>AVWL</sub>	2, 3	Address valid to WR low or RD low	125		4t <sub>CLCL</sub> -75		ns
t <sub>QVWX</sub>	2, 3	Data valid to WR transition	20		t <sub>CLCL</sub> -30		ns
t <sub>DW</sub>	2, 3	Data setup time before WR	220		7t <sub>CLCL</sub> -130		ns
t <sub>WHQX</sub>	2, 3	Data hold after WR	25		t <sub>CLCL</sub> -25		ns
t <sub>RLAZ</sub>	2, 3	RD low to address float		0		0	ns
t <sub>WHLH</sub>	2, 3	RD or WR high to ALE high	25	75	t <sub>CLCL</sub> -25	t <sub>CLCL</sub> +25	ns
<b>Shift Register</b>							
t <sub>XLXL</sub>	4	Serial port clock cycle time <sup>4</sup>	0.6		12t <sub>CLCL</sub>		μs
t <sub>QVXH</sub>	4	Output data setup to clock rising edge <sup>4</sup>	367		10t <sub>CLCL</sub> -133		ns
t <sub>XHQX</sub>	4	Output data hold after clock rising edge <sup>4</sup>	40		2t <sub>CLCL</sub> -60		ns
t <sub>XHDX</sub>	4	Input data hold after clock rising edge <sup>4</sup>	0		0		ns
t <sub>XHDV</sub>	4	Clock rising edge to input data valid <sup>4</sup>		367		10t <sub>CLCL</sub> -133	ns
<b>External Clock</b>							
t <sub>CHCX</sub>	5	High time <sup>4</sup>	17		17	t <sub>CLCL</sub> - t <sub>LOW</sub>	ns
t <sub>CLCX</sub>	5	Low time <sup>4</sup>	17		17	t <sub>CLCL</sub> - t <sub>HIGH</sub>	ns
t <sub>CLCH</sub>	5	Rise time <sup>4</sup>		20		20	ns
t <sub>CHCL</sub>	5	Fall time <sup>4</sup>		20		20	ns

## CMOS single-chip 8-bit microcontroller 87C652

### AC ELECTRICAL CHARACTERISTICS (Continued)

$T_{amb} = 0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$ , or  $T_{amb} = -40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $V_{SS} = 0\text{V}^{1,2}$

SYMBOL	PARAMETER	INPUT	OUTPUT
<b>I<sup>2</sup>C Interface</b>			
t <sub>HD</sub> : STA	START condition hold time	≥ 14 t <sub>CLCL</sub>	> 4.0μs <sup>4</sup>
t <sub>LOW</sub>	SCL low time	≥ 16 t <sub>CLCL</sub>	> 4.7μs <sup>4</sup>
t <sub>HIGH</sub>	SCL high time	≥ 14 t <sub>CLCL</sub>	> 4.0μs <sup>4</sup>
t <sub>RC</sub>	SCL rise time	≤ 1μs	– <sup>5</sup>
t <sub>FC</sub>	SCL fall time	≤ 0.3μs	< 0.3μs <sup>6</sup>
t <sub>SU</sub> : DAT1	Data set-up time	≥ 250ns	> 20 t <sub>CLCL</sub> – t <sub>RD</sub>
t <sub>SU</sub> : DAT2	SDA set-up time (before rep. START cond.)	≥ 250ns	> 1μs <sup>4</sup>
t <sub>SU</sub> : DAT3	SDA set-up time (before STOP cond.)	≥ 250ns	> 8 t <sub>CLCL</sub>
t <sub>HD</sub> : DAT	Data hold time	≥ 0ns	> 8 t <sub>CLCL</sub> – t <sub>FC</sub>
t <sub>SU</sub> : STA	Repeated START set-up time	≥ 14 t <sub>CLCL</sub> <sup>4</sup>	> 4.7μs <sup>4</sup>
t <sub>SU</sub> : STO	STOP condition set-up time	≥ 14 t <sub>CLCL</sub> <sup>4</sup>	> 4.0μs <sup>4</sup>
t <sub>BUF</sub>	Bus free time	≥ 14 t <sub>CLCL</sub> <sup>4</sup>	> 4.7μs <sup>4</sup>
t <sub>RD</sub>	SDA rise time	≤ 1μs <sup>7</sup>	– <sup>5</sup>
t <sub>FD</sub>	SDA fall time	≤ 300ns <sup>7</sup>	< 0.3μs <sup>6</sup>

NOTES:

- Parameters are valid over operating temperature range and voltage range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- These values are characterized but not 100% production tested.
- At 100 kbit/s. At other bit rates this value is inversely proportional to the bit-rate of 100 kbit/s.
- Determined by the external bus-line capacitance and the external bus-line pull-resistor, this must be < 1 $\mu$ s.
- Spikes on the SDA and SCL lines with a duration of less than 3 t<sub>CLCL</sub> will be filtered out. Maximum capacitance on bus-lines SDA and SCL = 400pF.
- t<sub>CLCL</sub> = 1/f<sub>OSC</sub> = one oscillator clock period at pin XTAL1. For 63ns < t<sub>CLCL</sub> < 285ns (16MHz > f<sub>OSC</sub> > 3.5MHz) the I<sup>2</sup>C interface meets the I<sup>2</sup>C-bus specification for bit-rates up to 100 kbit/s.

# CMOS single-chip 8-bit microcontroller 87C652

## EXPLANATION OF THE AC SYMBOLS

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

- A – Address
- C – Clock
- D – Input data
- H – Logic level high
- I – Instruction (program memory contents)
- L – Logic level low, or ALE
- P – PSEN

Q – Output data

R – RD signal

t – Time

V – Valid

W – WR signal

X – No longer a valid logic level

Z – Float

Examples:  $t_{AVLL}$  = Time for address valid to ALE low.

$t_{LLPL}$  = Time for ALE low to PSEN low.

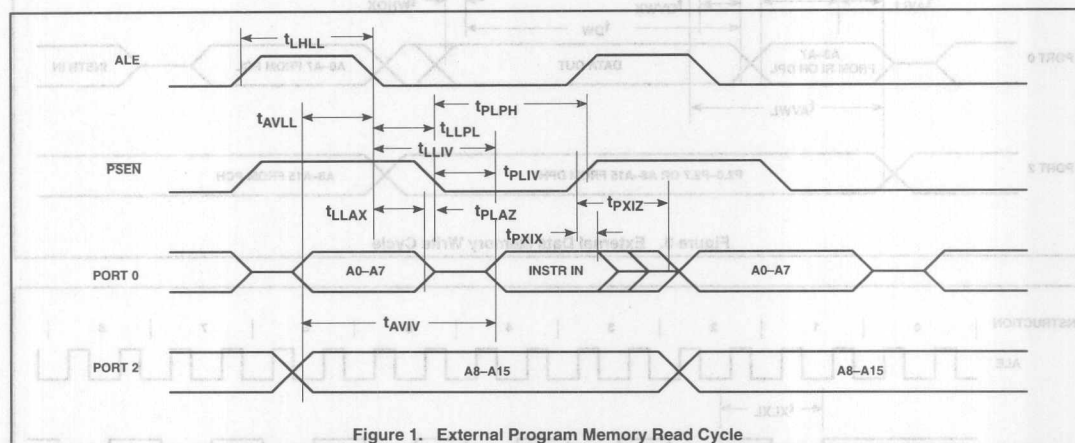


Figure 1. External Program Memory Read Cycle

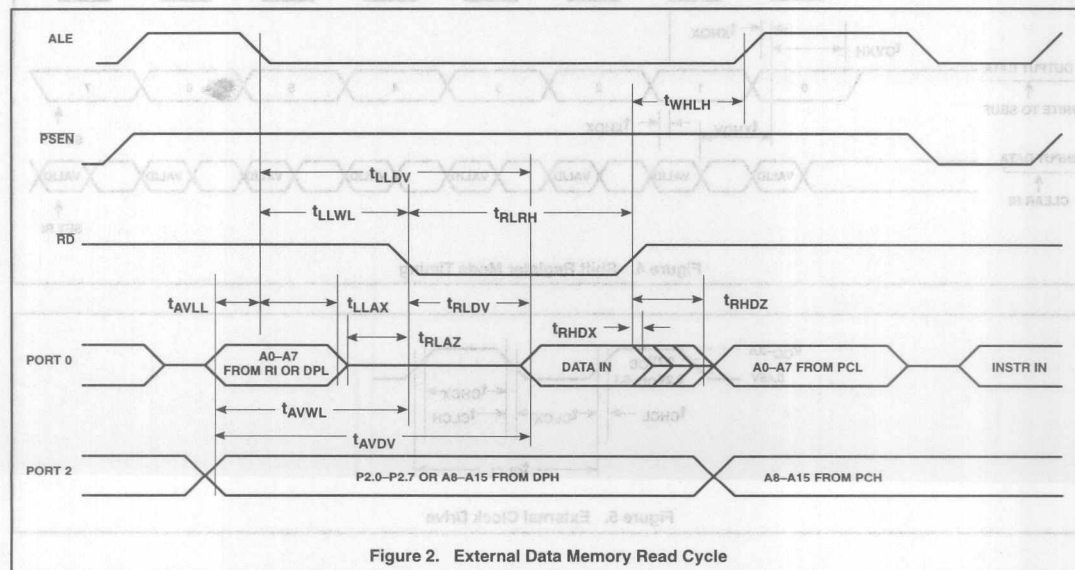


Figure 2. External Data Memory Read Cycle

## CMOS single-chip 8-bit microcontroller 87C652

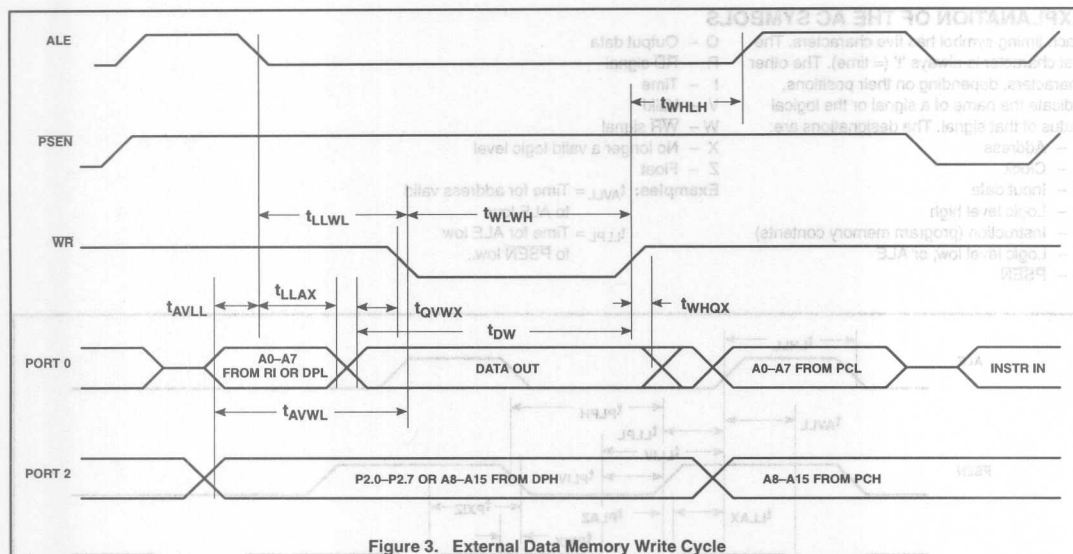


Figure 3. External Data Memory Write Cycle

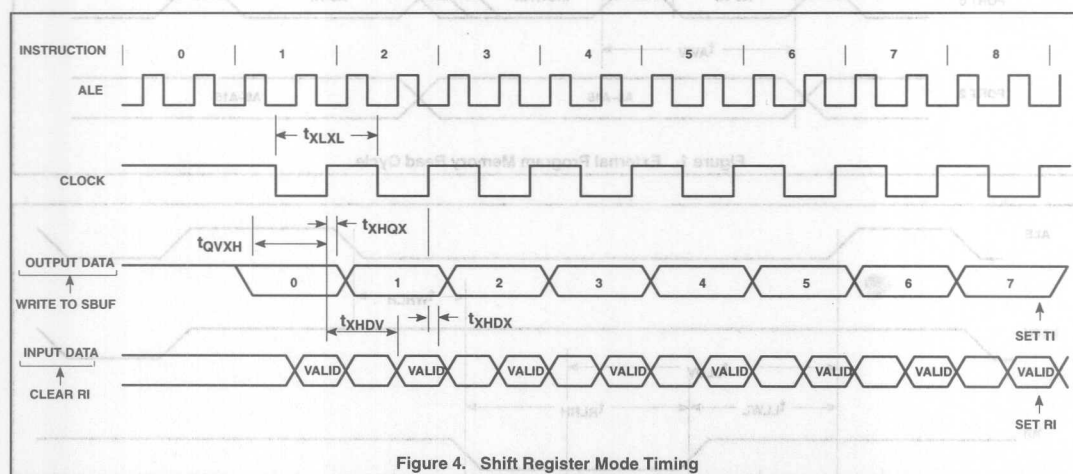


Figure 4. Shift Register Mode Timing

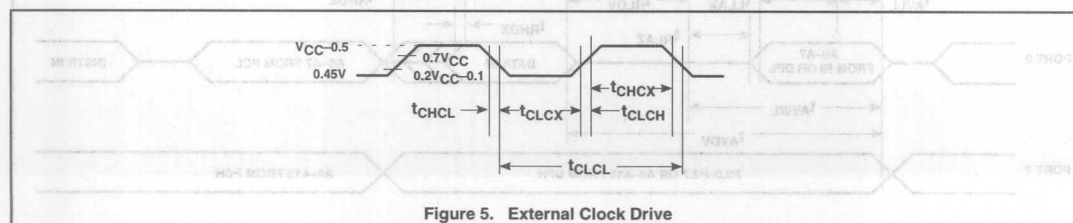


Figure 5. External Clock Drive

## CMOS single-chip 8-bit microcontroller 87C652

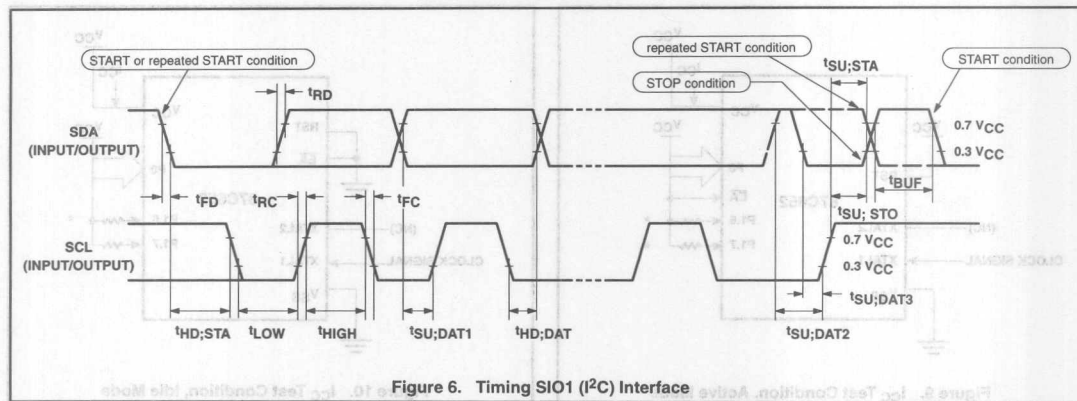


Figure 6. Timing SIO1 (I²C) Interface

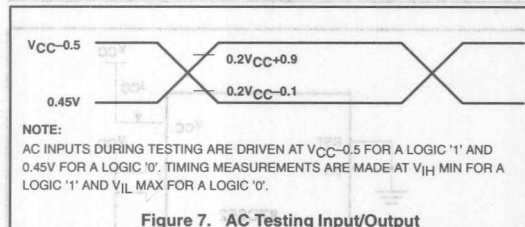


Figure 7. AC Testing Input/Output

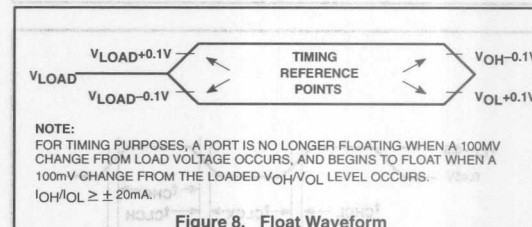
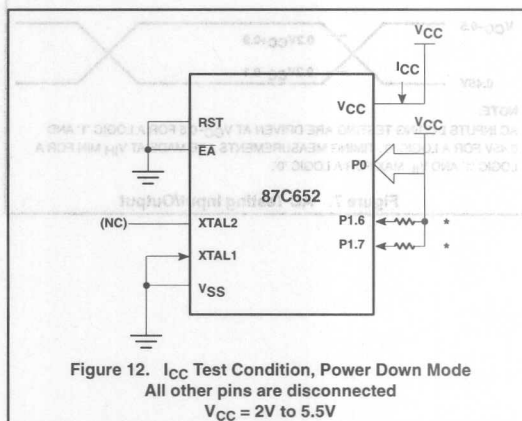
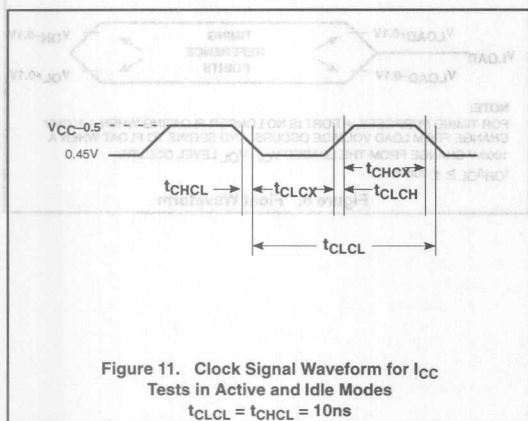
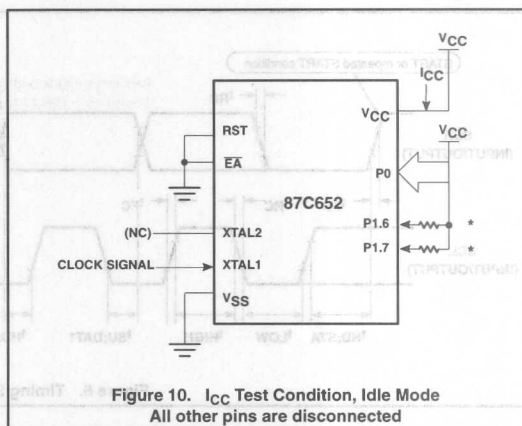
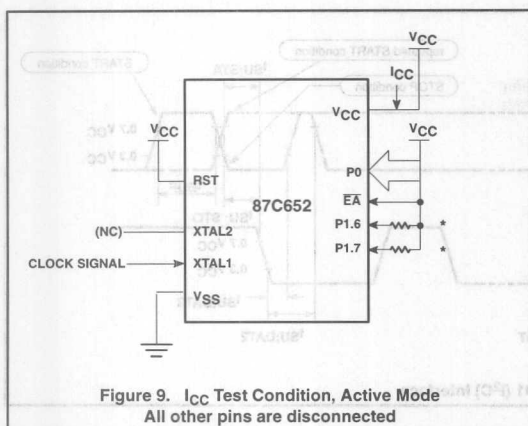


Figure 8. Float Waveform





**NOTE:**

- \* Ports 1.6 and 1.7 should be connected to  $V_{CC}$  through resistors of sufficiently high value such that the sink current into these pins does not exceed the  $I_{OL1}$  specification.

## CMOS single-chip 8-bit microcontroller 87C652

## EPROM CHARACTERISTICS

The 87C652 is programmed by using a modified Quick-Pulse Programming™ algorithm. It differs from older methods in the value used for  $V_{PP}$  (programming supply voltage) and in the width and number of the ALE/PROG pulses.

The 87C652 contains two signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes identify the device as an 87C652 manufactured by Philips Components.

Table 4 shows the logic levels for reading the signature byte, and for programming the program memory, the encryption table, and the lock bits. The circuit configuration and waveforms for quick-pulse programming are shown in Figures 13 and 14. Figure 15 shows the circuit configuration for normal program memory verification.

## Quick-Pulse Programming

The setup for microcontroller quick-pulse programming is shown in Figure 13. Note that the 87C652 is running with a 4 to 6MHz oscillator. The reason the oscillator needs to be running is that the device is executing internal address and program data transfers.

The address of the EPROM location to be programmed is applied to ports 1 and 2, as shown in Figure 13. The code byte to be programmed into that location is applied to port 0. RST, PSEN and pins of ports 2 and 3 specified in Table 4 are held at the 'Program Code Data' levels indicated in Table 4. The ALE/PROG is pulsed low 25 times as shown in Figure 14.

To program the encryption table, repeat the 25 pulse programming sequence for addresses 0 through 1FH, using the 'Pgm Encryption Table' levels. Do not forget that after the encryption table is programmed, verification cycles will produce only encrypted data.

To program the lock bits, repeat the 25 pulse programming sequence using the 'Pgm Lock Bit' levels. After one lock bit is programmed, further programming of the code memory and encryption table is disabled. However, the other lock bit can still be programmed.

Note that the  $\overline{EA}/V_{PP}$  pin must not be allowed to go above the maximum specified  $V_{PP}$  level for any amount of time. Even a narrow glitch above that voltage can cause permanent damage to the device. The  $V_{PP}$  source should be well regulated and free of glitches and overshoot.

## Program Verification

If lock bit 2 has not been programmed, the on-chip program memory can be read out for program verification. The address of the program memory locations to be read is applied to ports 1 and 2 as shown in Figure 15. The other pins are held at the 'Verify Code Data' levels indicated in Table 4. The contents of the address location will be emitted on port 0. External pull-ups are required on port 0 for this operation.

If the encryption table has been programmed, the data presented at port 0 will be the exclusive NOR of the program byte with one of the encryption bytes. The user will have to know the encryption table contents in order to correctly decode the verification data. The encryption table itself cannot be read out.

## Reading the Signature Bytes

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. The values are:

(030H) = 5H indicates manufactured by Philips

(031H) = 99H indicates 87C652

## Program/Verify Algorithms

Any algorithm in agreement with the conditions listed in Table 4, and which satisfies the timing specifications, is suitable.

## Erasure Characteristics

Erasure of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent erasure. **For this and secondary effects, it is recommended that an opaque label be placed over the window.** For elevated temperature or environments where solvents are being used, apply Kapton tape Fluorglas part number 2345-5, or equivalent.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 angstroms) to an integrated dose of at least 15W-sec/cm<sup>2</sup>. Exposing the EPROM to an ultraviolet lamp of 12,000uW/cm<sup>2</sup> rating for 20 to 39 minutes, at a distance of about 1 inch, should be sufficient. Erasure leaves the array in an all 1s state.

Table 4. EPROM Programming Modes

MODE	RST	PSEN	ALE/PROG	$\overline{EA}/V_{PP}$	P2.7	P2.6	P3.7	P3.6
Read signature	1	0	1	1	0	0	0	0
Program code data	1	0	0*	$V_{PP}$	1	0	1	1
Verify code data	1	0	1	1	0	0	1	1
Pgm encryption table	1	0	0*	$V_{PP}$	1	0	1	0
Pgm lock bit 1	1	0	0*	$V_{PP}$	1	1	1	1
Pgm lock bit 2	1	0	0*	$V_{PP}$	1	1	0	0

## NOTES:

1. '0' = Valid low for that pin, '1' = valid high for that pin.

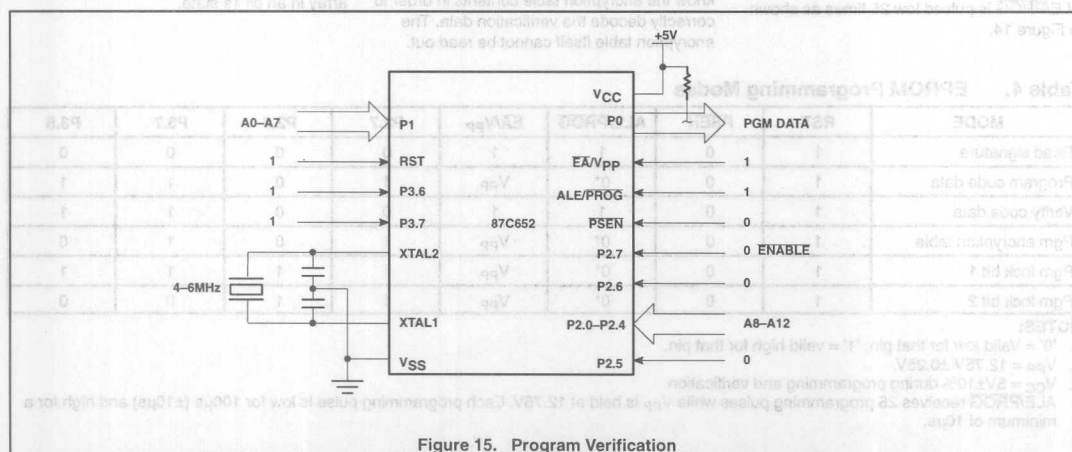
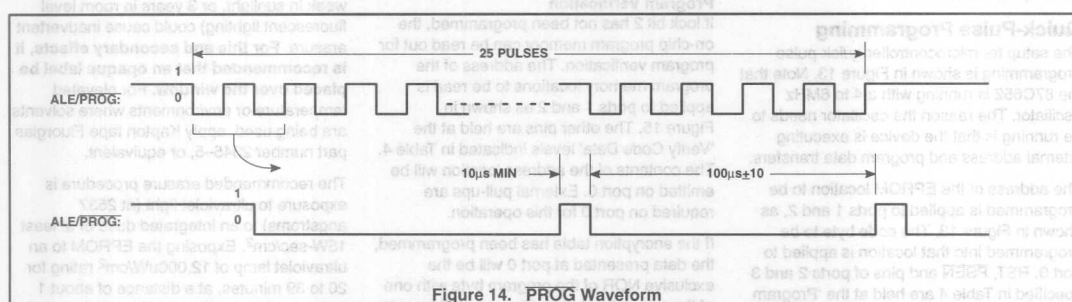
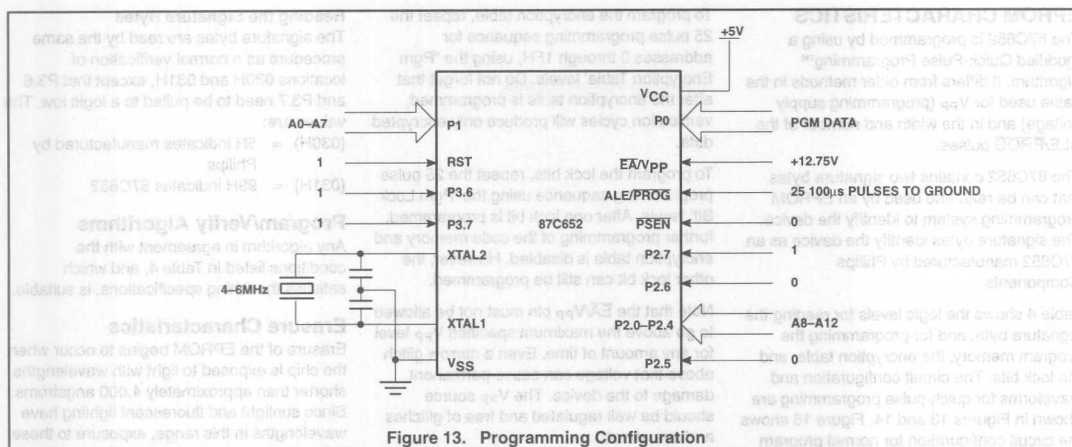
2.  $V_{PP} = 12.75V \pm 0.25V$ .

3.  $V_{CC} = 5V \pm 10\%$  during programming and verification.

\* ALE/PROG receives 25 programming pulses while  $V_{PP}$  is held at 12.75V. Each programming pulse is low for 100 $\mu$ s ( $\pm 10\mu$ s) and high for a minimum of 10 $\mu$ s.

™Trademark phrase of Intel Corporation.

## CMOS single-chip 8-bit microcontroller 87C652



## CMOS single-chip 8-bit microcontroller

87C652

## EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

 $T_{amb} = 21^{\circ}\text{C}$  to  $+27^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$  (see Figure 16)

SYMBOL	PARAMETER	MIN	MAX	UNIT
$V_{PP}$	Programming supply voltage	12.5	13.0	V
$I_{PP}$	Programming supply current		50	mA
$1/t_{CLCL}$	Oscillator frequency	4	6	MHz
$t_{AVGL}$	Address setup to PROG low*	$48t_{CLCL}$		
$t_{GHAX}$	Address hold after PROG*	$48t_{CLCL}$		
$t_{DVGL}$	Data setup to PROG low	$48t_{CLCL}$		
$t_{GHDX}$	Data hold after PROG	$48t_{CLCL}$		
$t_{EHS}$	P2.7 (ENABLE) high to $V_{PP}$	$48t_{CLCL}$		
$t_{SHGL}$	$V_{PP}$ setup to PROG low	10		$\mu\text{s}$
$t_{GHSL}$	$V_{PP}$ hold after PROG	10		$\mu\text{s}$
$t_{GLGH}$	PROG width	90		$\mu\text{s}$
$t_{AVQV}$	Address to data valid		$48t_{CLCL}$	
$t_{ELQZ}$	ENABLE low to data valid		$48t_{CLCL}$	
$t_{EHQZ}$	Data float after ENABLE	0	$48t_{CLCL}$	
$t_{GHGL}$	PROG high to PROG low	10		$\mu\text{s}$

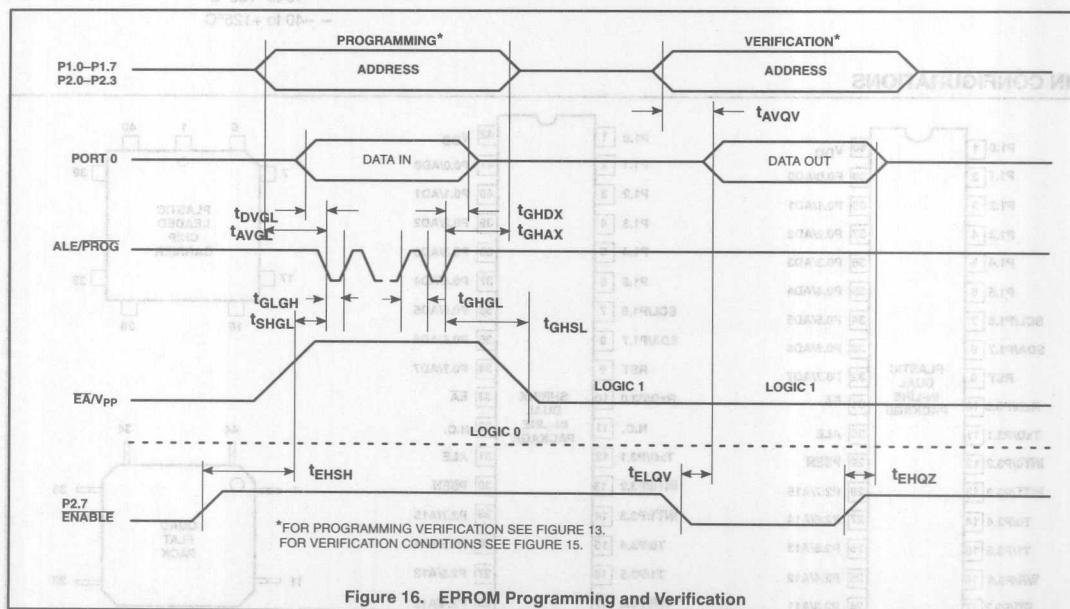


Figure 16. EPROM Programming and Verification



Purchase of Philips' I<sup>2</sup>C components conveys a license under the Philips' I<sup>2</sup>C patent to use the components in the I<sup>2</sup>C-system provided the system conforms to the I<sup>2</sup>C specifications defined by Philips.

# CMOS single-chip 8-bit microcontroller 83C654

## DESCRIPTION

The P83C654 Single-Chip 8-Bit Microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 83C654 has the same instruction set as the 80C51. Two versions of the derivative exist:

83C654 — 16k bytes mask programmable ROM

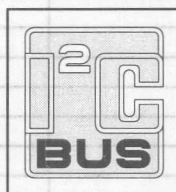
87C654 — EPROM version (described in a separate data sheet)

This device provides architectural enhancements that make it applicable in a variety of applications for general control systems. The 83C654 contains a non-volatile 16k × 8 read-only program memory, a volatile 256 × 8 read/write data memory, four 8-bit I/O ports, two 16-bit timer/event counters (identical to the timers of the 80C51), a multi-source, two-priority-level, nested interrupt structure, an I<sup>2</sup>C interface, UART and on-chip oscillator and timing circuits. For systems that require extra capability, the

83C654 can be expanded using standard TTL compatible memories and logic.

The device also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities.

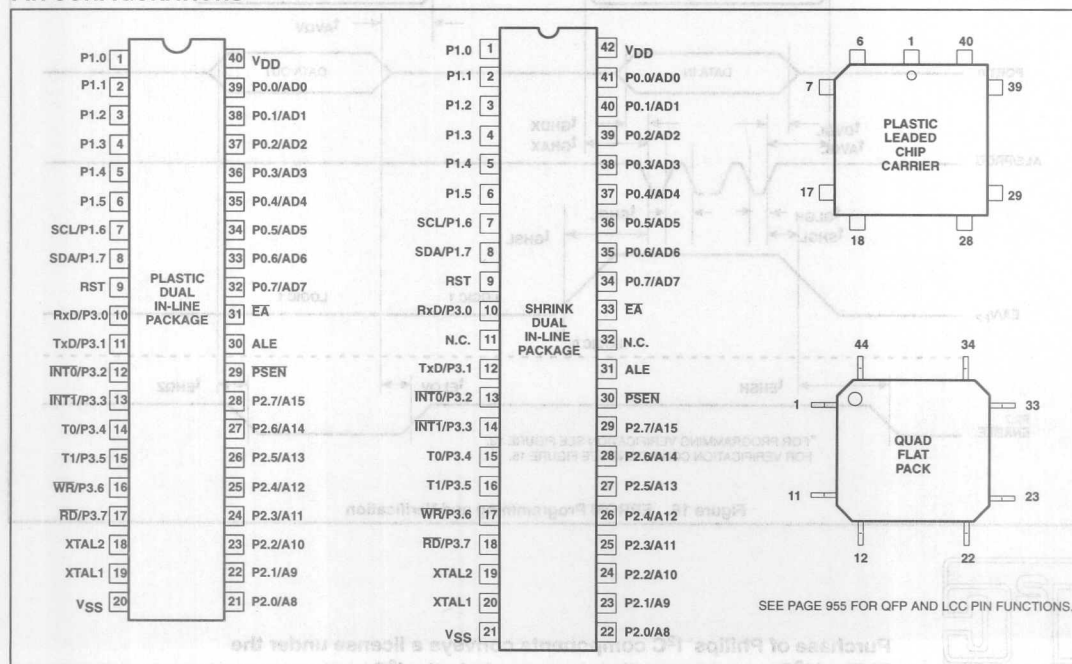
The instruction set consists of over 100 instructions: 49 one-byte, 45 two-byte and 17 three-byte. With a 16(24)MHz crystal, 58% of the instructions are executed in 0.75(0.5)μs and 40% in 1.5(1)μs. Multiply and divide instructions require 3(2)μs.



## FEATURES

- 80C51 central processing unit
- 16k × 8 ROM expandable externally to 64k bytes
- 256 × 8 RAM, expandable externally to 64k bytes
- Two standard 16-bit timer/counters
- Four 8-bit I/O ports
- I<sup>2</sup>C-bus serial I/O port with byte oriented master and slave functions
- Full-duplex UART facilities
- Power control modes
  - Idle mode
  - Power-down mode
- ROM code protection
- Extended frequency range: 1.2 to 24 MHz
- Three operating ambient temperature ranges:
  - 0 to +70°C
  - -40 to +85°C
  - -40 to +125°C

## PIN CONFIGURATIONS





## CMOS single-chip 8-bit microcontroller 83C654

## PLASTIC LEADED CHIP CARRIER

## PIN FUNCTIONS

Pin	Function	Pin	Function
1	NC	23	NC
2	P1.0	24	P2.0/A8
3	P1.1	25	P2.1/A9
4	P1.2	26	P2.2/A10
5	P1.3	27	P2.3/A11
6	P1.4	28	P2.4/A12
7	P1.5	29	P2.5/A13
8	P1.6/SCL	30	P2.6/A14
9	P1.7/SDA	31	P2.7/A15
10	RST	32	PSEN
11	P3.0/RxD	33	ALE
12	NC	34	NC
13	P3.1/TxD	35	EA
14	P3.2/INT0	36	P0.7/AD7
15	P3.3/INT1	37	P0.6/AD6
16	P3.4/T0	38	P0.5/AD5
17	P3.5/T1	39	P0.4/AD4
18	P3.6/WR	40	P0.3/AD3
19	P3.7/RD	41	P0.2/AD2
20	XTAL2	42	P0.1/AD1
21	XTAL1	43	P0.0/AD0
22	VSS	44	VDD

## PLASTIC QUAD FLAT PACK

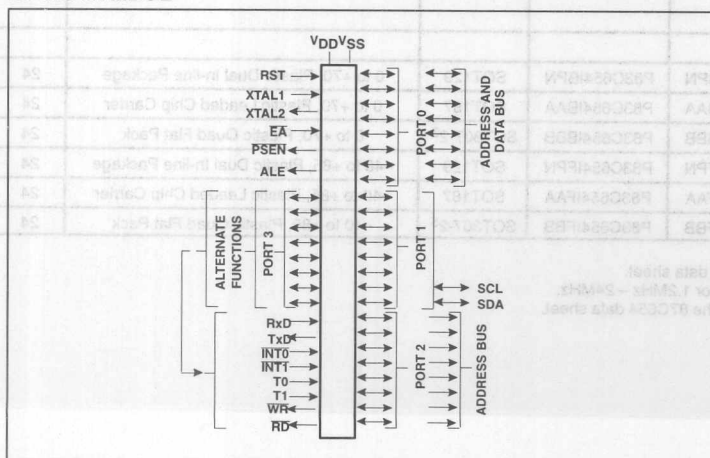
## PIN FUNCTIONS

Pin	Function	Pin	Function
1	P1.5	23	P2.5/A13
2	P1.6/SCL	24	P2.6/A14
3	P1.7/SDA	25	P2.7/A15
4	RST	26	PSEN
5	P3.0/RxD	27	ALE
6	VSS4	28	VSS2
7	P3.1/TxD	29	EA/Vpp
8	P3.2/INT0	30	P0.7/AD7
9	P3.3/INT1	31	P0.6/AD6
10	P3.4/T0	32	P0.5/AD5
11	P3.5/T1	33	P0.4/AD4
12	P3.6/WR	34	P0.3/AD3
13	P3.7/RD	35	P0.2/AD2
14	XTAL2	36	P0.1/AD1
15	XTAL1	37	P0.0/AD0
16	VSS1	38	VDD
17	NC	39	VSS3
18	P2.0/A8	40	P1.0
19	P2.1/A9	41	P1.1
20	P2.2/A10	42	P1.2
21	P2.3/A11	43	P1.3
22	P2.4/A12	44	P1.4

## NOTES TO QFP ONLY:

- Due to EMC improvements, all VSS pins (6, 16, 28, 39) must be connected to VSS on the 80C652/83C654.

## LOGIC SYMBOL



## CMOS single-chip 8-bit microcontroller

83C654

## ORDERING INFORMATION

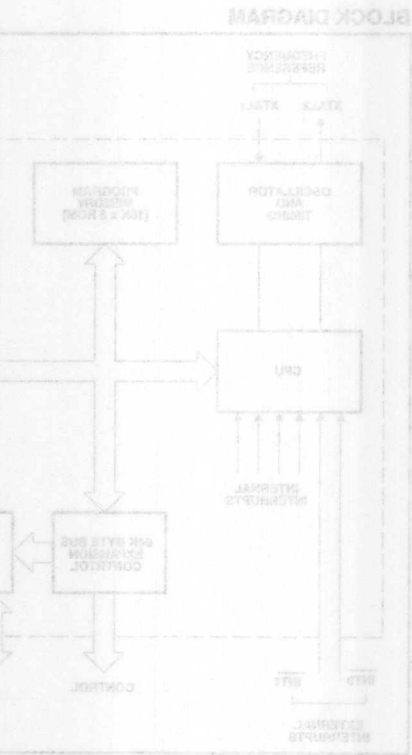
PHILIPS PART ORDER NUMBER PART MARKING		PHILIPS NORTH AMERICA PART ORDER NUMBER		Drawing Number	TEMPERATURE RANGE °C AND PACKAGE	FREQ MHz <sup>2,3</sup>
ROMless <sup>1</sup>	ROM	ROMless <sup>1</sup>	ROM			
P80C652FBP	P83C654FBP/xxx	P80C652FBPN	P83C654FBPN	SOT129	0 to +70, Plastic Dual In-line Package	16
					0 to +70, Ceramic Dual In-line Package w/Window	16
P80C652FBA	P83C654FBA/xxx	P80C652FBAA	P83C654FBAA	SOT187	0 to +70, Plastic Leaded Chip Carrier	16
P80C652FBB	P83C654FBB/xxx	P80C652FBBB	P83C654FBBB	SOT307-2 <sup>5</sup>	0 to +70, Plastic Quad Flat Pack	16
	P83C654FBB/xxx			SOT270	0 to +70, Plastic Shrink Dual In-Line Package	
P80C652FFP	P83C654FFP/xxx	P80C652FFPN	P83C654FFPN	SOT129	-40 to +85, Plastic Dual In-line Package	16
P80C652FFA	P83C654FFA/xxx	P80C652FFAA	P83C654FFAA	SOT187	-40 to +85, Plastic Leaded Chip Carrier	16
P80C652FFB	P83C654FFB/xxx	P80C652FFBB	P83C654FFBB	SOT307-2 <sup>5</sup>	-40 to +85, Plastic Quad Flat Pack	16
P80C652FHP	P83C654FHP/xxx	P80C652FHPN	P83C654FHPN	SOT129	-40 to +125, Plastic Dual In-line Package	16
P80C652FHA	P83C654FHA/xxx	P80C652FHAA	P83C654FHAA	SOT187	-40 to +125, Plastic Leaded Chip Carrier	16
P80C652FHB	P83C654FHB/xxx	P80C652FHBB	P83C654FHBB	SOT307-2 <sup>5</sup>	-40 to +125, Plastic Quad Flat Pack	16
P80C652IBP	P83C654IBP/xxx	P80C652IBPN	P83C654IBPN	SOT129	0 to +70, Plastic Dual In-line Package	24
P80C652IBA	P83C654IBA/xxx	P80C652IBAA	P83C654IBAA	SOT187	0 to +70, Plastic Leaded Chip Carrier	24
P80C652IBB	P83C654IBB/xxx	P80C652IBBB	P83C654IBBB	SOT307-2 <sup>5</sup>	0 to +70, Plastic Quad Flat Pack	24
P80C652IFP	P83C654IFP/xxx	P80C652IFPN	P83C654IFPN	SOT129	-40 to +85, Plastic Dual In-line Package	24
P80C652IFA	P83C654IFA/xxx	P80C652IFAA	P83C654IFAA	SOT187	-40 to +85, Plastic Leaded Chip Carrier	24
P80C652IFB	P83C654IFB/xxx	P80C652IFBB	P83C654IFBB	SOT307-2 <sup>5</sup>	-40 to +85, Plastic Quad Flat Pack	24

NOTES:

1. For full specification, see the 80C652/83C652 data sheet.
2. 83C654 frequency range is 1.2MHz – 16MHz or 1.2MHz – 24MHz.
3. For specification of the EPROM version, see the 87C654 data sheet.
4. xxx denotes the ROM code number.
5. SOT311 replaced by SOT307-2.

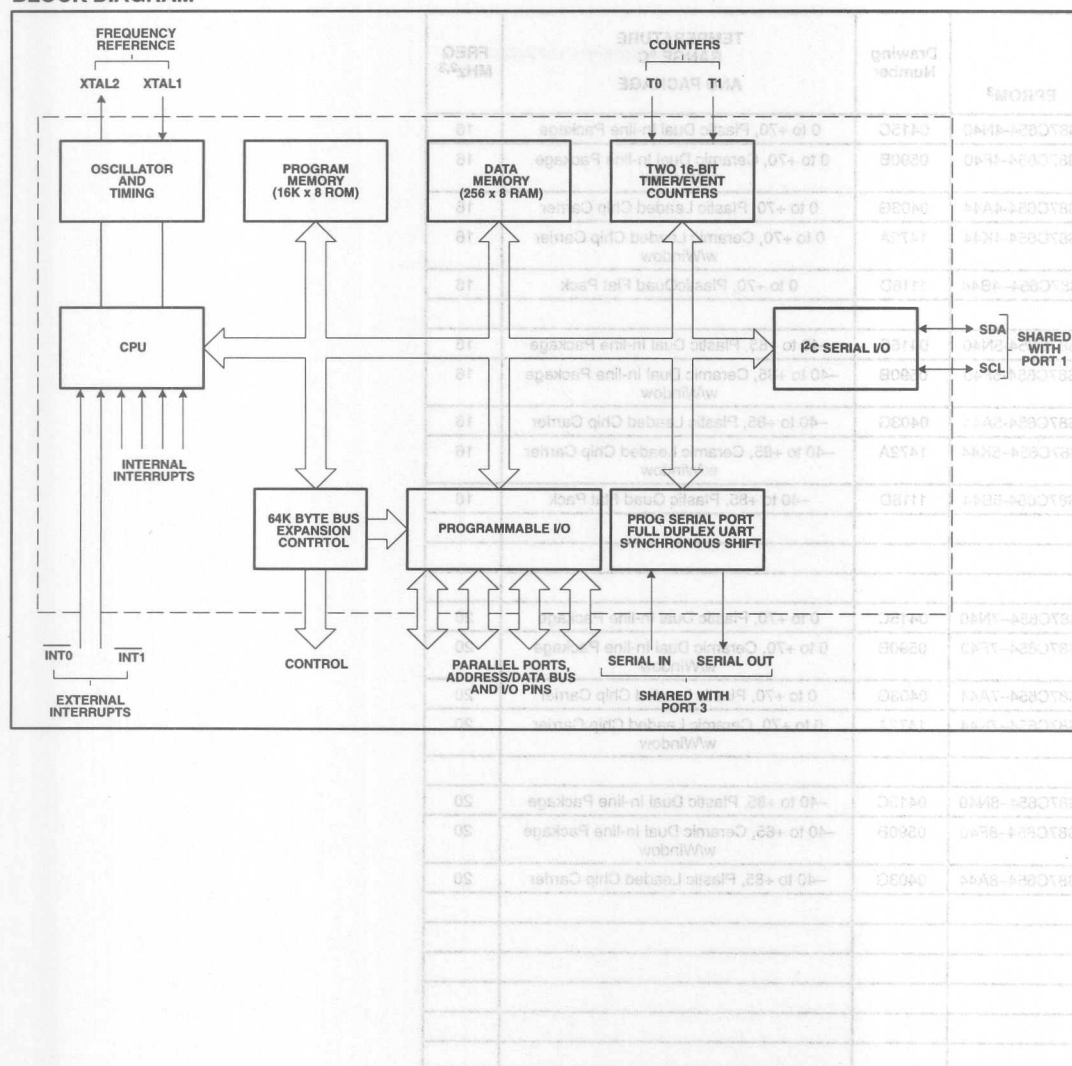
CMOS single-chip 8-bit microcontroller 83C654

EPROM <sup>3</sup>	Drawing Number	TEMPERATURE RANGE °C AND PACKAGE	FREQ MHz <sup>2,3</sup>
S87C654-4N40	0415C	0 to +70, Plastic Dual In-line Package	16
S87C654-4F40	0590B	0 to +70, Ceramic Dual In-line Package w/Window	16
S87C654-4A44	0403G	0 to +70, Plastic Leaded Chip Carrier	16
S87C654-4K44	1472A	0 to +70, Ceramic Leaded Chip Carrier w/Window	16
S87C654-4B44	1118D	0 to +70, Plastic Quad Flat Pack	16
S87C654-5N40	0415C	-40 to +85, Plastic Dual In-line Package	16
S87C654-5F40	0590B	-40 to +85, Ceramic Dual In-line Package w/Window	16
S87C654-5A44	0403G	-40 to +85, Plastic Leaded Chip Carrier	16
S87C654-5K44	1472A	-40 to +85, Ceramic Leaded Chip Carrier w/Window	16
S87C654-5B44	1118D	-40 to +85, Plastic Quad Flat Pack	16
S87C654-7N40	0415C	0 to +70, Plastic Dual In-line Package	20
S87C654-7F40	0590B	0 to +70, Ceramic Dual In-line Package w/Window	20
S87C654-7A44	0403G	0 to +70, Plastic Leaded Chip Carrier	20
S87C654-7K44	1472A	0 to +70, Ceramic Leaded Chip Carrier w/Window	20
S87C654-8N40	0415C	-40 to +85, Plastic Dual In-line Package	20
S87C654-8F40	0590B	-40 to +85, Ceramic Dual In-line Package w/Window	20
S87C654-8A44	0403G	-40 to +85, Plastic Leaded Chip Carrier	20



## CMOS single-chip 8-bit microcontroller 83C654

### BLOCK DIAGRAM



## CMOS single-chip 8-bit microcontroller 83C654

## PIN DESCRIPTIONS

MNEMONIC	PIN NUMBER			TYPE	NAME AND FUNCTION
	DIP	PLCC	QFP		
V <sub>SS</sub>	20	22	6, 16, 28, 39	I	<b>Ground:</b> 0V reference. With the QFP package all V <sub>SS</sub> pins (V <sub>SS1</sub> to V <sub>SS4</sub> ) must be connected.
V <sub>DD</sub>	40	44	38	I	<b>Power Supply:</b> This is the power supply voltage for normal, idle, and power-down operation.
P0.0–0.7	39–32	43–36	37–30	I/O	<b>Port 0:</b> Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s.
P1.0–P1.7	1–8	2–9	40–44, 1–3	I/O	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pull-ups, except P1.6 and P1.7 which are open drain. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Alternate functions include: <b>SCL:</b> I <sup>2</sup> C-bus serial port clock line. <b>SDA:</b> I <sup>2</sup> C-bus serial port data line.
P1.6	7	8	2	I/O	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register.
P1.7	8	9	3	I/O	
P2.0–P2.7	21–28	24–31	18–25	I/O	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 3 also serves the special features of the 80C51 family, as listed below: <b>RxD (P3.0):</b> Serial input port <b>TxD (P3.1):</b> Serial output port <b>INT0 (P3.2):</b> External interrupt <b>INT1 (P3.3):</b> External interrupt <b>T0 (P3.4):</b> Timer 0 external input <b>T1 (P3.5):</b> Timer 1 external input <b>WR (P3.6):</b> External data memory write strobe <b>RD (P3.7):</b> External data memory read strobe
P3.0–P3.7	10–17	11, 13–19	5, 7–13	I/O	
	10	11	5	I	<b>Reset:</b> A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to V <sub>SS</sub> permits a power-on reset using only an external capacitor to V <sub>DD</sub> .
	11	13	7	O	
	12	14	8	I	
	13	15	9	I	
	14	16	10	I	
	15	17	11	I	
	16	18	12	O	
	17	19	13	O	
RST	9	10	4	I	
ALE	30	33	27	I/O	<b>Address Latch Enable:</b> Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency. Note that one ALE pulse is skipped during each access to external data memory.
PSEN	29	32	26	O	<b>Program Store Enable:</b> Read strobe to external program memory via Port 0 and Port 2. It is activated twice each machine cycle during fetches from the external program memory. When executing out of external program memory two activations of PSEN are skipped during each access to external data memory. PSEN is not activated (remains HIGH) during no fetches from external program memory. PSEN can sink/source 8 LSTTL inputs and can drive CMOS inputs without external pull-ups.
EA	31	35	29	I	<b>External Access:</b> If during a RESET, EA is held at TTL, level HIGH, the CPU executes out of the internal program memory ROM provided the Program Counter is less than 16384. If during a RESET, EA is held a TTL LOW level, the CPU executes out of external program memory. EA is not allowed to float.
XTAL1	19	21	15	I	<b>Crystal 1:</b> Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	20	14	O	<b>Crystal 2:</b> Output from the inverting oscillator amplifier.

## NOTE:

To avoid "latch-up" effect at power-on, the voltage on any pin at any time must not be higher than V<sub>DD</sub> + 0.5V or V<sub>SS</sub> – 0.5V, respectively.



## CMOS single-chip 8-bit microcontroller 83C654

Table 1. 8XC652/654 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE	
			MSB								LSB	
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H	
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H	
DPTR:	Data pointer (2 bytes)											
DPH	Data pointer high	83H									00H	
DPL	Data pointer low	82H									00H	
			AF	AE	AD	AC	AB	AA	A9	A8		
IE*#	Interrupt enable	A8H	EA		ES1	ES0	ET1	EX1	ET0	EX0	0x000000B	
			BF	BE	BD	BC	BB	BA	B9	B8		
IP*#	Interrupt priority	B8H	—		PS1	PS0	PT1	PX1	PT0	PX0	xx000000B	
			87	86	85	84	83	82	81	80		
P0*	Port 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH	
			97	96	95	94	93	92	91	90		
P1*#	Port 1	90H	SDA	SCL							FFH	
			A7	A6	A5	A4	A3	A2	A1	A0		
P2*	Port 2	A0H	A15	A14	A13	A12	A11	A10	A9	A8	FFH	
			B7	B6	B5	B4	B3	B2	B1	B0		
P3*	Port 3	B0H	RD	WR	T1	T0	INT1	INT0	TXD	RXD	FFH	
PCON	Power control	87H	SMOD	—	—	—	GF1	GF0	PD	IDL	0xxx0000B	
			9F	9E	9D	9C	9B	9A	99	98		
SOCON*#	Serial 0 port control	98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00H	
S0BUF#	Serial 0 data buffer	99H									xxxxxxxB	
			D7	D6	D5	D4	D3	D2	D1	D0		
PSW*	Program status word	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	00H	
S1DAT#	Serial 1 data	DAH									00H	
SP	Stack pointer	81H									07H	
S1ADR#	Serial 1 address	DBH	SLAVE ADDRESS								GC	00H
S1STA#	Serial 1 status	D9H	SC4	SC3	SC2	SC1	SC0	0	0	0	F8H	
			DF	DE	DD	DC	DB	DA	D9	D8		
S1CON*#	Serial 1 control	D8H	CR2	ENS1	STA	STO	SI	AA	CR1	CR0	00000000B	
			8F	8E	8D	8C	8B	8A	89	88		
TCON*	Timer control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H	
TH1	Timer high 1	8DH									00H	
TH0	Timer high 0	8CH									00H	
TL1	Timer low 1	8BH									00H	
TH0	Timer low 0	8AH									00H	
TMOD	Timer mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H	

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.

## CMOS single-chip 8-bit microcontroller 83C654

**ROM CODE PROTECTION  
(83C654)**

The 83C654 has an additional security feature. ROM code protection may be selected by setting a mask-programmable security bit (i.e., user dependent). This feature may be requested during ROM code submission. When selected, the ROM code is protected and cannot be read out at any time by any test mode or by any instruction in the external program memory space.

The MOVC instructions are the only instructions that have access to program code in the internal or external program memory. The EA input is latched during RESET and is "don't care" after RESET (also if the security bit is not set). This implementation prevents reading internal program code by switching from external program memory to internal program memory during a MOVC instruction or any other instruction that uses immediate data.

**OSCILLATOR  
CHARACTERISTICS**

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator, as shown in the Logic Symbol, page 955.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

**Reset**

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. To insure a good power-on reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few milliseconds) plus two machine cycles. At power-on, the voltage on V<sub>DD</sub> and RST must come up at the same time for a proper start-up.

**Idle Mode**

In the idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle mode can be terminated either by any

enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

**Power-Down Mode**

In the power-down mode, the oscillator is stopped and the instruction to invoke power-down is the last instruction executed. Only the contents of the on-chip RAM are preserved. A hardware reset is the only way to terminate the power-down mode. The control bits for the reduced power modes are in the special function register PCON. Table 2 shows the state of the I/O ports during low current operating modes.

**I<sup>2</sup>C SERIAL COMMUNICATION —  
SIO1**

The I<sup>2</sup>C serial port is identical to the I<sup>2</sup>C serial port on the 8XC552. The operation of this subsystem is described in detail in the 8XC552 section of this manual.

Note that in both the 8XC652/4 and the 8XC552 the I<sup>2</sup>C pins are alternate functions to port pins P1.6 and P1.7. Because of this, P1.6 and P1.7 on these parts do not have a pull-up structure as found on the 80C51. Therefore P1.6 and P1.7 have open drain outputs on the 8XC652/4.

**Table 2. External Pin Status During Idle and Power-Down Mode**

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

**Serial Control Register (S1CON) – See Table 3**

S1CON (D8H)	CR2	ENS1	STA	STO	SI	AA	CR1	CR0
-------------	-----	------	-----	-----	----	----	-----	-----

Bits CR0, CR1 and CR2 determine the serial clock frequency that is generated in the master mode of operation.

**Table 3. Serial Clock Rates**

CR2	CR1	CR0	BIT FREQUENCY (kHz) AT f <sub>OSC</sub>				f <sub>OSC</sub> DIVIDED BY
			6MHz	12MHz	16MHz	24MHz	
0	0	0	23	47	62.5	94	256
0	0	1	27	54	71	107 <sup>1</sup>	224
0	1	0	31.25	62.5	83.3	125 <sup>1</sup>	192
0	1	1	37	75	100	150 <sup>1</sup>	160
1	0	0	6.25	12.5	17	25	960
1	0	1	50	100	133 <sup>1</sup>	200 <sup>1</sup>	120
1	1	0	100	200 <sup>1</sup>	267 <sup>1</sup>	400 <sup>1</sup>	60
1	1	1	0.24 < 62.5 0 to 255	0.49 < 62.5 0 to 254	0.65 < 55.6 0 to 253	0.98 < 50.0 0 to 251	96 × (256 – (reload value Timer 1)) reload value range Timer 1 (in mode 2)

**NOTES:**

1. These frequencies exceed the upper limit of 100kHz of the I<sup>2</sup>C-bus specification and cannot be used in an I<sup>2</sup>C-bus application.

## ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>

PARAMETER	RATING	UNIT
Storage temperature range	-65 to +150	°C
Voltage on any other pin to V <sub>SS</sub>	-0.5 to +6.5	V
Input, output current on any single pin	±5	mA
Power dissipation (based on package heat transfer limitations, not device power consumption)	1	W

### NOTES:

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
- This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
- Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V<sub>SS</sub> unless otherwise noted.

## DEVICE SPECIFICATIONS

TYPE	SUPPLY VOLTAGE (V)		FREQUENCY (MHz)		TEMPERATURE RANGE (°C)
	MIN.	MAX.	MIN.	MAX.	
P83C654FBx	4.0	6.0	1.2	16	0 to +70
P83C654FFx	4.0	6.0	1.2	16	-40 to +85
P83C654FHx	4.5	5.5	1.2	16	-40 to +125
P83C654IBx	4.5	5.5	1.2	24	0 to +70
P83C654IFx	4.5	5.5	1.2	24	-40 to +85

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Flash	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Flash	Data	Data	Data

Serial Control Register (SICR) - See Table 3

SICR (bits)	CR2	SRST	STA	STO	SI	AA	CR1	CR0
-------------	-----	------	-----	-----	----	----	-----	-----

Bits CR0, CR1 and CR2 determine the serial clock frequency that is generated in the master mode of operation.

Table 3. Serial Clock Rates

f <sub>clock</sub> DIVIDED BY	BIT FREQUENCY (MHz) AT f <sub>clock</sub>				CR0	CR1	CR2
	24MHz	16MHz	12MHz	8MHz			
256	94	62.5	47	33	0	0	0
128	187	125	94	66	1	0	0
64	374	250	187	132	0	1	0
32	748	500	374	264	1	1	0
16	1496	1000	748	528	0	0	1
8	2992	2000	1496	1056	1	0	1
4	5984	4000	2992	2112	0	1	1
2	11968	8000	5984	4224	1	1	1

reduced value range Timer 1 (in mode 2)  
reduced value range Timer 1 (in mode 2)  
reduced value range Timer 1 (in mode 2)

NOTES:  
1. These frequencies exceed the upper limit of 100MHz of the f<sub>clock</sub> bus specification and cannot be used in f<sub>clock</sub> bus applications.

## CMOS single-chip 8-bit microcontroller 83C654

## DC ELECTRICAL CHARACTERISTICS

 $V_{SS} = 0V$ 

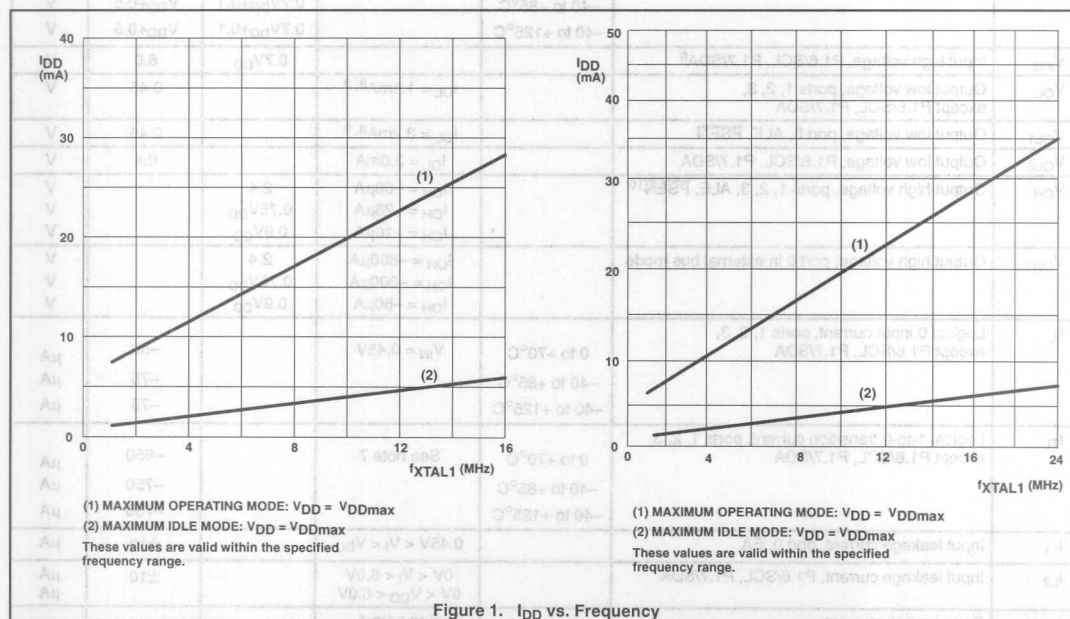
SYMBOL	PARAMETER	PART TYPE	TEST CONDITIONS	LIMITS		UNIT
				MIN.	MAX.	
$V_{IL}$	Input low voltage, except EA, P1.6/SCL, P1.7/SDA	0 to +70°C -40 to +85°C -40 to +125°C		-0.5 -0.5 -0.5	$0.2V_{DD}-0.1$ $0.2V_{DD}-0.15$ $0.2V_{DD}-0.25$	V V V
$V_{IL1}$	Input low voltage to EA	0 to +70°C -40 to +85°C -40 to +125°C		-0.5 -0.5 -0.5	$0.2V_{DD}-0.3$ $0.2V_{DD}-0.35$ $0.2V_{DD}-0.45$	V V V
$V_{IL2}$	Input low voltage to P1.6/SCL, P1.7/SDA <sup>6</sup>			-0.5	$0.3V_{DD}$	V
$V_{IH}$	Input high voltage, except XTAL1, RST, P1.6/SCL, P1.7/SDA	0 to +70°C -40 to +85°C -40 to +125°C		$0.2V_{DD}+0.9$ $0.2V_{DD}+1.0$ $0.2V_{DD}+1.0$	$V_{DD}+0.5$ $V_{DD}+0.5$ $V_{DD}+0.5$	V V V
$V_{IH1}$	Input high voltage, XTAL1, RST	0 to +70°C -40 to +85°C -40 to +125°C		$0.7V_{DD}$ $0.7V_{DD}+0.1$ $0.7V_{DD}+0.1$	$V_{DD}+0.5$ $V_{DD}+0.5$ $V_{DD}+0.5$	V V V
$V_{IH2}$	Input high voltage, P1.6/SCL, P1.7/SDA <sup>6</sup>			$0.7V_{DD}$	6.0	V
$V_{OL}$	Output low voltage, ports 1, 2, 3, except P1.6/SCL, P1.7/SDA		$I_{OL} = 1.6mA^{8,9}$		0.45	V
$V_{OL1}$	Output low voltage, port 0, ALE, PSEN		$I_{OL} = 3.2mA^{8,9}$		0.45	V
$V_{OL2}$	Output low voltage, P1.6/SCL, P1.7/SDA		$I_{OL} = 3.0mA$		0.4	V
$V_{OH}$	Output high voltage, ports 1, 2, 3, ALE, PSEN <sup>10</sup>		$I_{OH} = -60\mu A$ $I_{OH} = -25\mu A$ $I_{OH} = -10\mu A$	2.4 $0.75V_{DD}$ $0.9V_{DD}$		V V V
$V_{OH1}$	Output high voltage; port 0 in external bus mode		$I_{OH} = -800\mu A$ $I_{OH} = -300\mu A$ $I_{OH} = -80\mu A$	2.4 $0.75V_{DD}$ $0.9V_{DD}$		V V V
$I_{IL}$	Logical 0 input current, ports 1, 2, 3, except P1.6/SCL, P1.7/SDA	0 to +70°C -40 to +85°C -40 to +125°C	$V_{IN} = 0.45V$		-50 -75 -75	$\mu A$ $\mu A$ $\mu A$
$I_{TL}$	Logical 1-to-0 transition current, ports 1, 2, 3, except P1.6/SCL, P1.7/SDA	0 to +70°C -40 to +85°C -40 to +125°C	See note 7		-650 -750 -750	$\mu A$ $\mu A$ $\mu A$
$I_{L1}$	Input leakage current, port 0, EA		$0.45V < V_I < V_{DD}$		$\pm 10$	$\mu A$
$I_{L2}$	Input leakage current, P1.6/SCL, P1.7/SDA		$0V < V_I < 6.0V$ $0V < V_{DD} < 6.0V$		$\pm 10$	$\mu A$ $\mu A$
$I_{DD}$	Power supply current: Active mode @ 16MHz <sup>2, 11</sup> Active mode @ 24MHz <sup>2, 11</sup> Idle mode @ 16MHz <sup>3, 11</sup> Idle mode @ 24MHz <sup>3, 11</sup> Power down mode <sup>4, 5</sup> Power down mode <sup>4, 5</sup>	-40 to +125°C	See note 1 $V_{DD}=6.0V$ $V_{DD}=5.5V$		28.0 35.0 6 7 50 100	mA mA mA mA $\mu A$ $\mu A$
$R_{RST}$	Internal reset pull-down resistor			50	150	k $\Omega$
$C_{IO}$	Pin capacitance		Freq.=1MHz		10	pF

NOTES ON NEXT PAGE.

## CMOS single-chip 8-bit microcontroller 83C654

## NOTES FOR DC ELECTRICAL CHARACTERISTICS:

- See Figures 9 through 11 for  $I_{DD}$  test conditions.
- The operating supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 5\text{ns}$ ;  $V_{IL} = V_{SS} + 0.5\text{V}$ ;  $V_{IH} = V_{DD} - 0.5\text{V}$ ; XTAL2 not connected; EA = RST = Port 0 = P1.6 = P1.7 =  $V_{DD}$ . See Figure 9.
- The idle mode supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 5\text{ns}$ ;  $V_{IL} = V_{SS} + 0.5\text{V}$ ;  $V_{IH} = V_{DD} - 0.5\text{V}$ ; XTAL2 not connected; Port 0 = P1.6 = P1.7 =  $V_{DD}$ ; EA = RST =  $V_{SS}$ . See Figure 10.
- The power-down current is measured with all output pins disconnected; XTAL2 not connected; Port 0 = P1.6 = P1.7 =  $V_{DD}$ ; EA = RST =  $V_{SS}$ . See Figure 11.
- $2\text{V} \leq V_{DD} \leq V_{DD\text{max}}$ .
- The input threshold voltage of P1.6 and P1.7 (SIO1) meets the I<sup>2</sup>C specification, so an input voltage below  $0.3V_{DD}$  will be recognized as a logic 0 while an input voltage above  $0.7V_{DD}$  will be recognized as a logic 1.
- Pins of ports 1, 2, and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when  $V_{IN}$  is approximately 2V.
- Capacitive loading on ports 0 and 2 may cause spurious noise to be superimposed on the  $V_{OL}$ s of ALE and ports 1 and 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100pF), the noise pulse on the ALE pin may exceed 0.8V. In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.
- Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows: Maximum  $I_{OL} = 10\text{mA}$  per port pin; Maximum  $I_{OL} = 26\text{mA}$  total for Port 0; Maximum  $I_{OL} = 15\text{mA}$  total for Ports 1, 2, and 3; Maximum  $I_{OL} = 71\text{mA}$  total for all output pins. If  $I_{OL}$  exceeds the test conditions,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
- Capacitive loading on ports 0 and 2 may cause the  $V_{OH}$  on ALE and PSEN to momentarily fall below the  $0.9V_{DD}$  specification when the address bits are stabilizing.
- $I_{DD\text{MAX}}$  for other frequencies can be derived from Figure 1, where FREQ is the external oscillator frequency in MHz.  $I_{DD\text{MAX}}$  is given in mA.

Figure 1.  $I_{DD}$  vs. Frequency



## CMOS single-chip 8-bit microcontroller 83C654

AC ELECTRICAL CHARACTERISTICS<sup>1, 2</sup> (16 MHz type)

SYMBOL	FIGURE	PARAMETER	16MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
$1/t_{CLCL}$	2	Oscillator frequency			1.2	16	MHz
$t_{LHL}$	2	ALE pulse width	85		$2t_{CLCL}-40$		ns
$t_{AVLL}$	2	Address valid to ALE low	8		$t_{CLCL}-55$		ns
$t_{LLAX}$	2	Address hold after ALE low	28		$t_{CLCL}-35$		ns
$t_{LLIV}$	2	ALE low to valid instruction in		150	$4t_{CLCL}-100$		ns
$t_{LLPL}$	2	ALE low to PSEN low	23		$t_{CLCL}-40$		ns
$t_{PLPH}$	2	PSEN pulse width	143		$3t_{CLCL}-45$		ns
$t_{PLIV}$	2	PSEN low to valid instruction in		83	$3t_{CLCL}-105$		ns
$t_{PXIX}$	2	Input instruction hold after PSEN	0		0		ns
$t_{PXIZ}$	2	Input instruction float after PSEN		38	$t_{CLCL}-25$		ns
$t_{AVIV}$	2	Address to valid instruction in		208	$5t_{CLCL}-105$		ns
$t_{PLAZ}$	2	PSEN low to address float		10	10		ns
<b>Data Memory</b>							
$t_{RLRH}$	3, 4	RD pulse width	275		$6t_{CLCL}-100$		ns
$t_{WLWH}$	3, 4	WR pulse width	275		$6t_{CLCL}-100$		ns
$t_{RLDV}$	3, 4	RD low to valid data in		148	$5t_{CLCL}-165$		ns
$t_{RHDX}$	3, 4	Data hold after RD	0		0		ns
$t_{RHDZ}$	3, 4	Data float after RD		55	$2t_{CLCL}-70$		ns
$t_{LLDV}$	3, 4	ALE low to valid data in		350	$8t_{CLCL}-150$		ns
$t_{AVDV}$	3, 4	Address to valid data in		398	$9t_{CLCL}-165$		ns
$t_{LLWL}$	3, 4	ALE low to RD or WR low	138	238	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
$t_{AVWL}$	3, 4	Address valid to WR low or RD low	120		$4t_{CLCL}-130$		ns
$t_{QVWX}$	3, 4	Data valid to WR transition	3		$t_{CLCL}-60$		ns
$t_{DW}$	3, 4	Data setup time before WR	288		$7t_{CLCL}-150$		ns
$t_{WHQX}$	3, 4	Data hold after WR	13		$t_{CLCL}-50$		ns
$t_{RLAZ}$	3, 4	RD low to address float		0	0		ns
$t_{WHLH}$	3, 4	RD or WR high to ALE high	23	103	$t_{CLCL}-40$	$t_{CLCL}+40$	ns
<b>Shift Register</b>							
$t_{XLXL}$	5	Serial port clock cycle time <sup>3</sup>	0.75		$12t_{CLCL}$		$\mu$ s
$t_{QVXH}$	5	Output data setup to clock rising edge <sup>3</sup>	492		$10t_{CLCL}-133$		ns
$t_{XHDX}$	5	Output data hold after clock rising edge <sup>3</sup>	80		$2t_{CLCL}-117$		ns
$t_{XHDX}$	5	Input data hold after clock rising edge <sup>3</sup>	0		0		ns
$t_{XHDV}$	5	Clock rising edge to input data valid <sup>3</sup>		492	$10t_{CLCL}-133$		ns
<b>External Clock</b>							
$t_{CHCX}$	6	High time <sup>3</sup>	20		20	$t_{CLCL} - t_{CLCX}$	ns
$t_{CLCX}$	6	Low time <sup>3</sup>	20		20	$t_{CLCL} - t_{CHCX}$	ns
$t_{CLCH}$	6	Rise time <sup>3</sup>		20	20		ns
$t_{CHCL}$	6	Fall time <sup>3</sup>		20	20		ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- These values are characterized but not 100% production tested.

# AC ELECTRICAL CHARACTERISTICS<sup>1, 2</sup> (24 MHz type)

SYMBOL	FIGURE	PARAMETER	24MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
$1/t_{CLCL}$	2	Oscillator frequency			1.2	24	MHz
$t_{LHL}$	2	ALE pulse width	43		$2t_{CLCL}-40$		ns
$t_{AVLL}$	2	Address valid to ALE low	17		$t_{CLCL}-25$		ns
$t_{LLAX}$	2	Address hold after ALE low	17		$t_{CLCL}-25$		ns
$t_{LLIV}$	2	ALE low to valid instruction in		102		$4t_{CLCL}-65$	ns
$t_{LLPL}$	2	ALE low to PSEN low	17		$t_{CLCL}-25$		ns
$t_{PLPH}$	2	PSEN pulse width	80		$3t_{CLCL}-45$		ns
$t_{PLIV}$	2	PSEN low to valid instruction in		65		$3t_{CLCL}-60$	ns
$t_{PXIX}$	2	Input instruction hold after PSEN	0		0		ns
$t_{PXIZ}$	2	Input instruction float after PSEN		17		$t_{CLCL}-25$	ns
$t_{AVIV}$	2	Address to valid instruction in		128		$5t_{CLCL}-80$	ns
$t_{PLAZ}$	2	PSEN low to address float		10		10	ns
<b>Data Memory</b>							
$t_{RLRH}$	3, 4	RD pulse width	150		$6t_{CLCL}-100$		ns
$t_{WLWH}$	3, 4	WR pulse width	150		$6t_{CLCL}-100$		ns
$t_{RLDV}$	3, 4	RD low to valid data in		118		$5t_{CLCL}-90$	ns
$t_{RHDZ}$	3, 4	Data hold after RD	0		0		ns
$t_{RHDZ}$	3, 4	Data float after RD		55		$2t_{CLCL}-28$	ns
$t_{LLDV}$	3, 4	ALE low to valid data in		180		$8t_{CLCL}-150$	ns
$t_{AVDV}$	3, 4	Address to valid data in		210		$9t_{CLCL}-165$	ns
$t_{LLWL}$	3, 4	ALE low to RD or WR low	75	175	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
$t_{AVWL}$	3, 4	Address valid to WR low or RD low	92		$4t_{CLCL}-75$		ns
$t_{QVWX}$	3, 4	Data valid to WR transition	12		$t_{CLCL}-30$		ns
$t_{DW}$	3, 4	Data setup time before WR	162		$7t_{CLCL}-130$		ns
$t_{WHQX}$	3, 4	Data hold after WR	17		$t_{CLCL}-25$		ns
$t_{RLAZ}$	3, 4	RD low to address float		0		0	ns
$t_{WHLH}$	3, 4	RD or WR high to ALE high	17	67	$t_{CLCL}-25$	$t_{CLCL}+25$	ns
<b>Shift Register</b>							
$t_{XLXL}$	5	Serial port clock cycle time <sup>3</sup>	0.5		$12t_{CLCL}$		μs
$t_{QVXH}$	5	Output data setup to clock rising edge <sup>3</sup>	283		$10t_{CLCL}-133$		ns
$t_{XHGX}$	5	Output data hold after clock rising edge <sup>3</sup>	23		$2t_{CLCL}-60$		ns
$t_{XHDX}$	5	Input data hold after clock rising edge <sup>3</sup>	0		0		ns
$t_{XHDV}$	5	Clock rising edge to input data valid <sup>3</sup>		283		$10t_{CLCL}-133$	ns
<b>External Clock</b>							
$t_{CHCX}$	6	High time <sup>3</sup>	17		17	$t_{CLCL} - t_{CLCX}$	ns
$t_{CLCX}$	6	Low time <sup>3</sup>	17		17	$t_{CLCL} - t_{CHCX}$	ns
$t_{CLCH}$	6	Rise time <sup>3</sup>		5		5	ns
$t_{CHCL}$	6	Fall time <sup>3</sup>		5		5	ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- These values are characterized but not 100% production tested.

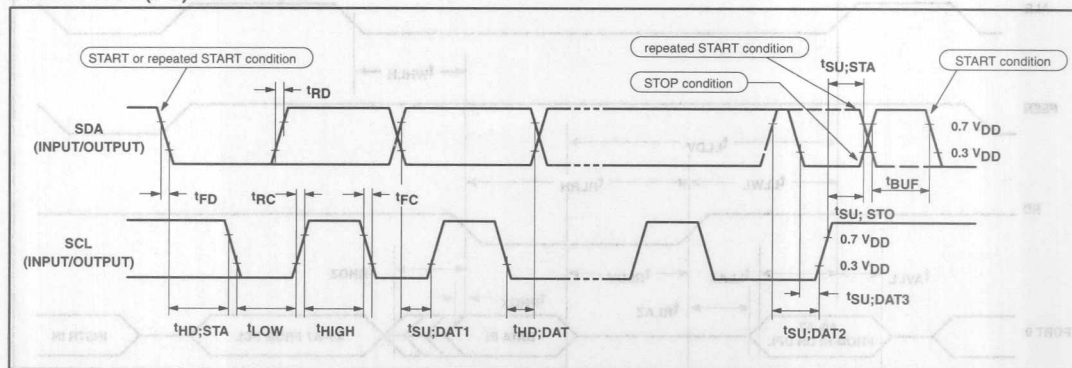
## CMOS single-chip 8-bit microcontroller 83C654

AC ELECTRICAL CHARACTERISTICS – I<sup>2</sup>C INTERFACE

SYMBOL	PARAMETER	INPUT	OUTPUT
<b>SCL TIMING CHARACTERISTICS</b>			
$t_{HD; STA}$	START condition hold time	$\geq 14 t_{CLCL}$	$> 4.0\mu s^1$
$t_{LOW}$	SCL LOW time	$\geq 16 t_{CLCL}$	$> 4.7\mu s^1$
$t_{HIGH}$	SCL HIGH time	$\geq 14 t_{CLCL}$	$> 4.0\mu s^1$
$t_{RC}$	SCL rise time	$\leq 1\mu s$	- 2
$t_{FC}$	SCL fall time	$\leq 0.3\mu s$	$< 0.3\mu s^3$
<b>SDA TIMING CHARACTERISTICS</b>			
$t_{SU; DAT1}$	Data set-up time	$\geq 250ns$	$> 20 t_{CLCL} - t_{RD}$
$t_{SU; DAT2}$	SDA set-up time (before rep. START cond.)	$\geq 250ns$	$> 1\mu s^1$
$t_{SU; DAT3}$	SDA set-up time (before STOP cond.)	$\geq 250ns$	$> 8 t_{CLCL}$
$t_{HD; DAT}$	Data hold time	$\geq 0ns$	$> 8 t_{CLCL} - t_{FC}$
$t_{SU; STA}$	Repeated START set-up time	$\geq 14 t_{CLCL}$	$> 4.7\mu s^1$
$t_{SU; STO}$	STOP condition set-up time	$\geq 14 t_{CLCL}$	$> 4.0\mu s^1$
$t_{BUF}$	Bus free time	$\geq 14 t_{CLCL}$	$> 4.7\mu s^1$
$t_{RD}$	SDA rise time	$\leq 1\mu s$	- 2
$t_{FD}$	SDA fall time	$\leq 0.3\mu s$	$< 0.3\mu s^3$

## NOTES:

1. At 100 kbit/s. At other bit rates this value is inversely proportional to the bit-rate of 100 kbit/s.
2. Determined by the external bus-line capacitance and the external bus-line pull-resistor, this must be  $< 1\mu s$ .
3. Spikes on the SDA and SCL lines with a duration of less than  $3 t_{CLCL}$  will be filtered out. Maximum capacitance on bus-lines SDA and SCL = 400pF.
4.  $t_{CLCL} = 1/f_{OSC}$  = one oscillator clock period at pin XTAL1. For  $63ns (42ns) < t_{CLCL} < 285ns (16MHz (24MHz)) > f_{OSC} > 3.5MHz$  the SiO1 interface meets the I<sup>2</sup>C-bus specification for bit-rates up to 100 kbit/s.

TIMING SIO1 (I<sup>2</sup>C) INTERFACE

## CMOS single-chip 8-bit microcontroller 83C654

### EXPLANATION OF THE AC SYMBOLS

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

- A – Address
- C – Clock
- D – Input data
- H – Logic level high
- I – Instruction (program memory contents)
- L – Logic level low, or ALE
- P – PSEN

Q – Output data

R -  $\overline{RD}$  signal

t -

Y - Valid

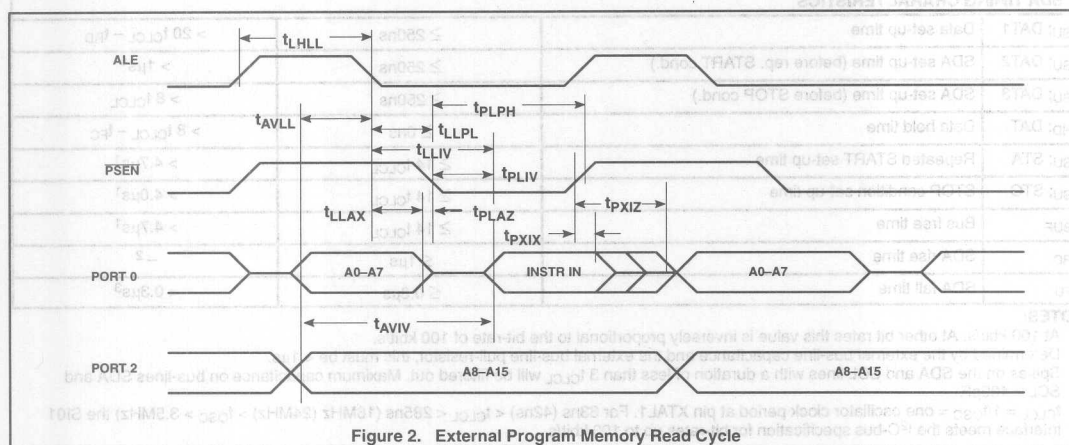
W –  $\overline{WR}$  signal

X – No longer a valid logic level

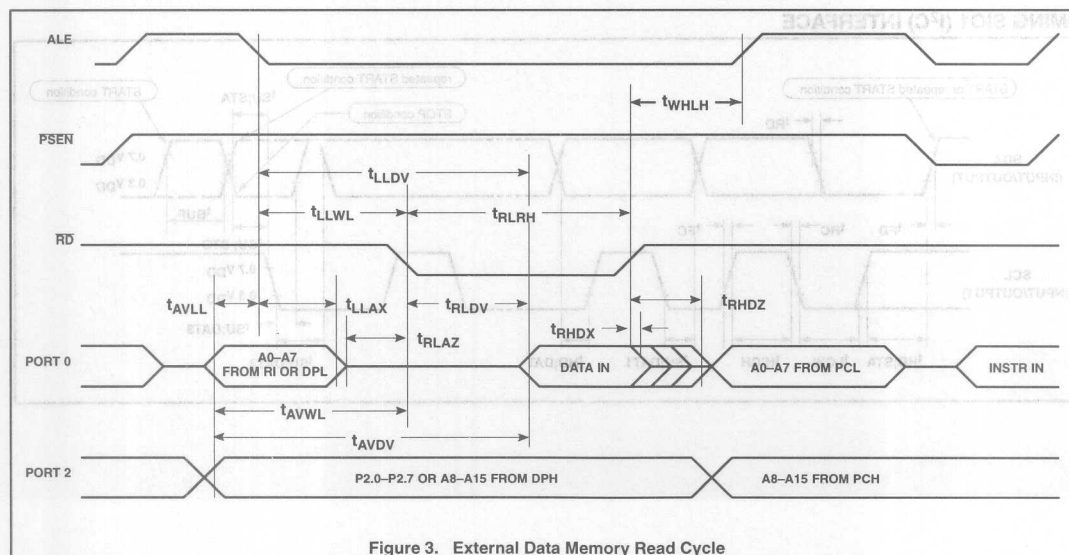
Z - Float

**Examples:**  $t_{AVLL}$  = Time for address valid to ALE low.

$t_{LLPL}$  = Time for ALE low to  $\overline{PSEN}$  low.



### Figure 2. External Program Memory Read Cycle



### Figure 3. External Data Memory Read Cycle

## CMOS single-chip 8-bit microcontroller 83C654

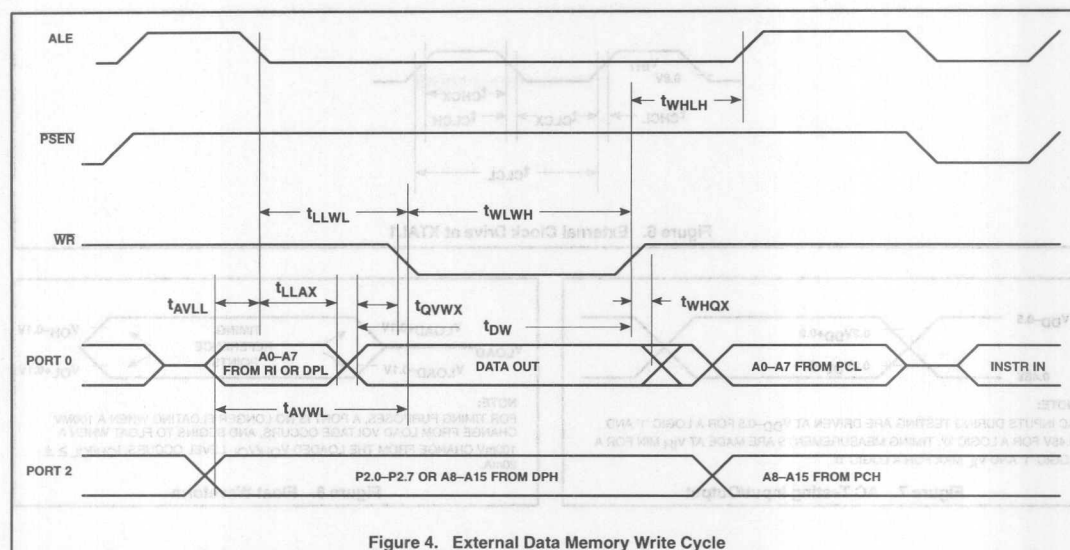


Figure 4. External Data Memory Write Cycle

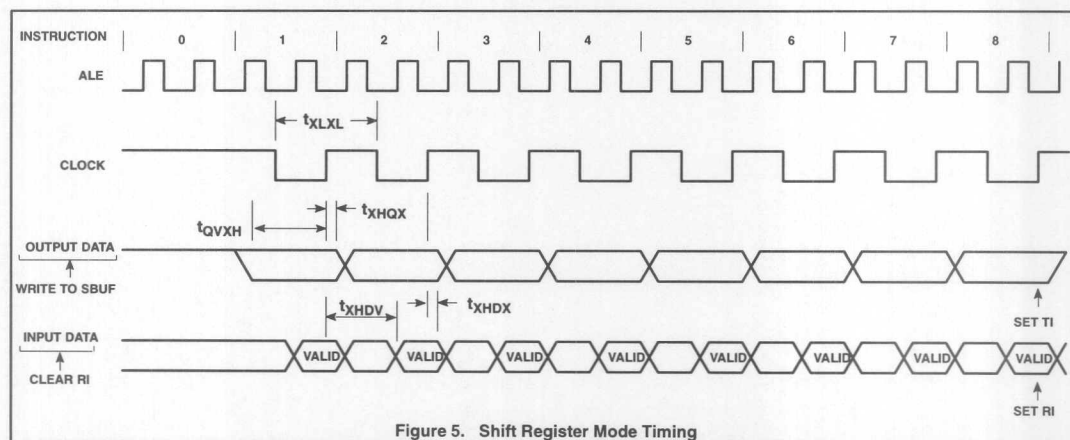
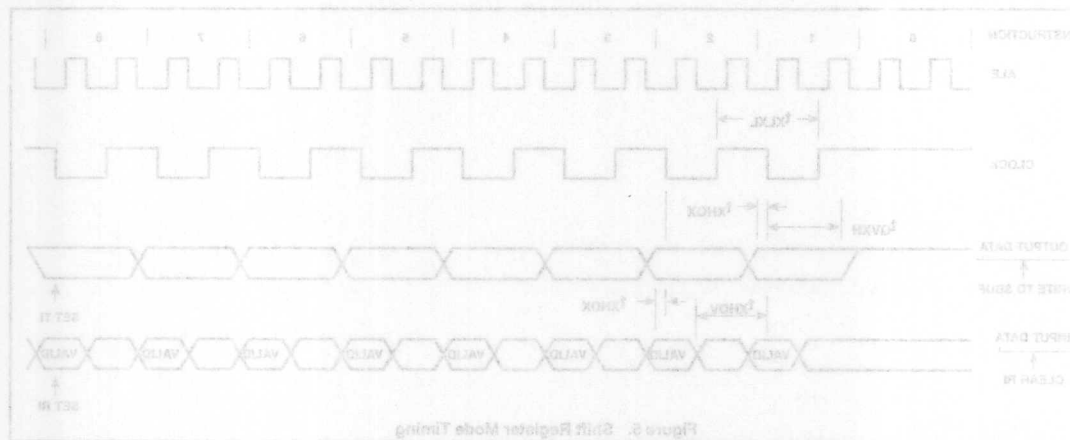
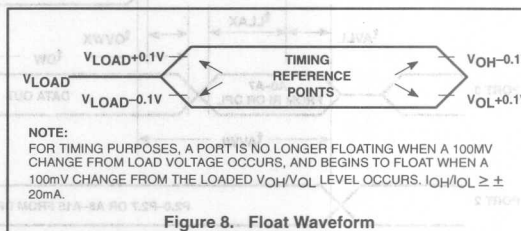
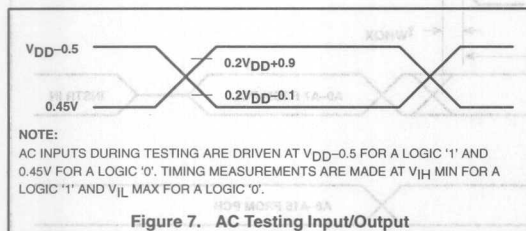
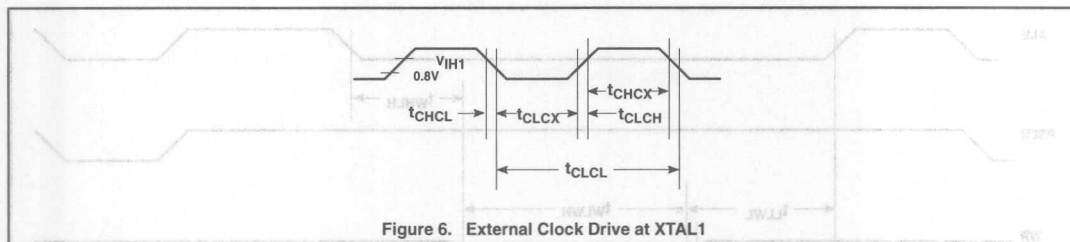


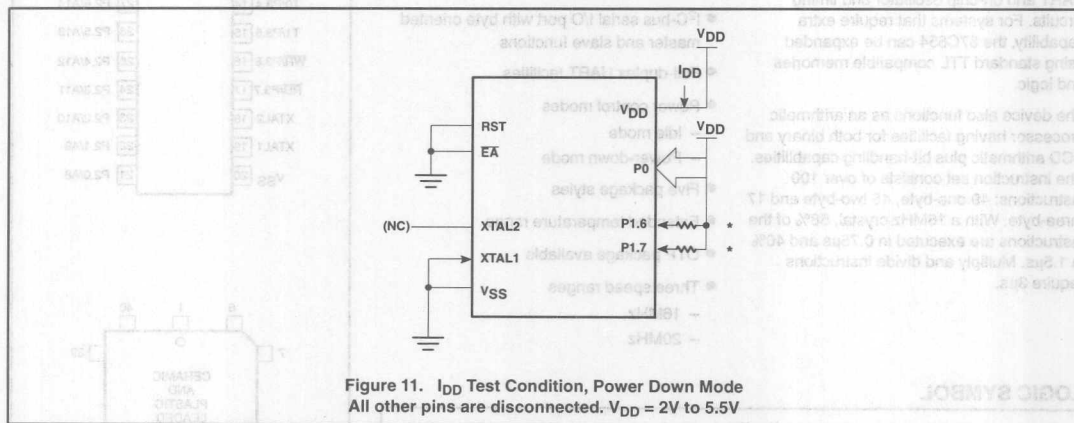
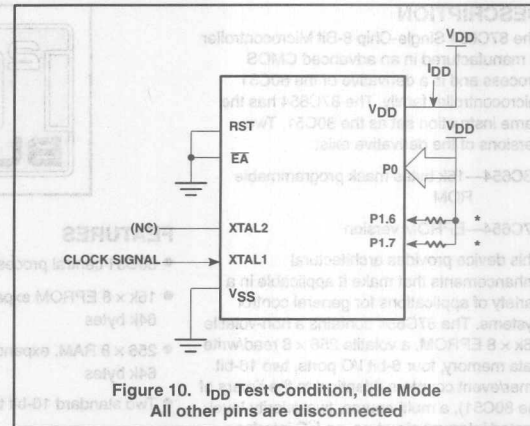
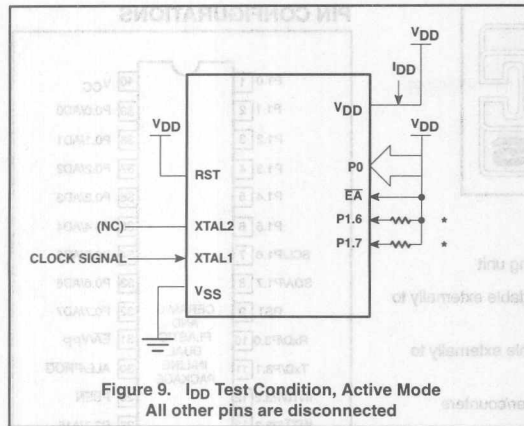
Figure 5. Shift Register Mode Timing



## CMOS single-chip 8-bit microcontroller 83C654



## CMOS single-chip 8-bit microcontroller 83C654

**NOTE:**

- \* Ports 1.6 and 1.7 should be connected to  $V_{CC}$  through resistors of sufficiently high value such that the sink current into these pins does not exceed the  $I_{OL1}$  specification.



Purchase of Philips' I<sup>2</sup>C components conveys a license under the Philips' I<sup>2</sup>C patent to use the components in the I<sup>2</sup>C-system provided the system conforms to the I<sup>2</sup>C specifications defined by Philips.

# CMOS single-chip 8-bit microcontroller 87C654

## DESCRIPTION

The 87C654 Single-Chip 8-Bit Microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 87C654 has the same instruction set as the 80C51. Two versions of the derivative exist:

83C654—16k bytes mask programmable ROM

87C654—EPROM version

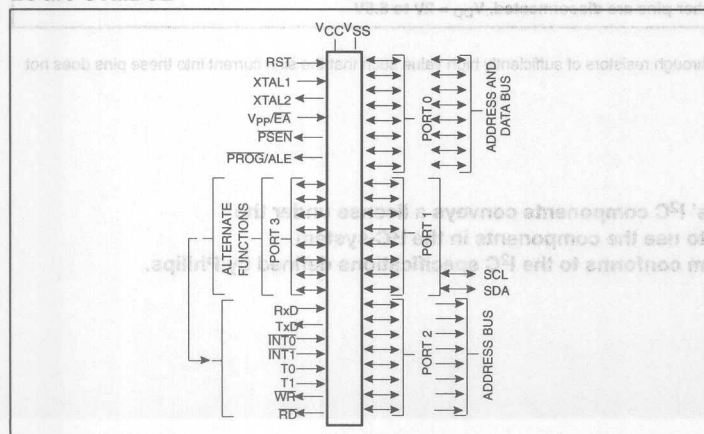
This device provides architectural enhancements that make it applicable in a variety of applications for general control systems. The 87C654 contains a non-volatile 16k × 8 EPROM, a volatile 256 × 8 read/write data memory, four 8-bit I/O ports, two 16-bit timer/event counters (identical to the timers of the 80C51), a multi-source, two-priority-level, nested interrupt structure, an I<sup>2</sup>C interface, UART and on-chip oscillator and timing circuits. For systems that require extra capability, the 87C654 can be expanded using standard TTL compatible memories and logic.

The device also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of over 100 instructions: 49 one-byte, 45 two-byte and 17 three-byte. With a 16MHz crystal, 58% of the instructions are executed in 0.75μs and 40% in 1.5μs. Multiply and divide instructions require 3μs.

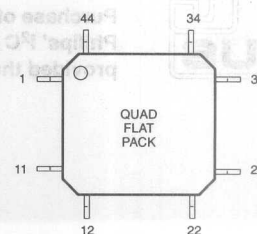
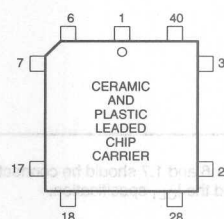
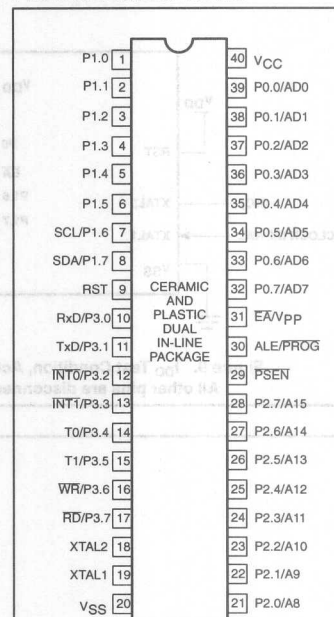
## FEATURES

- 80C51 central processing unit
- 16k × 8 EPROM expandable externally to 64k bytes
- 256 × 8 RAM, expandable externally to 64k bytes
- Two standard 16-bit timer/counters
- Four 8-bit I/O ports
- I<sup>2</sup>C-bus serial I/O port with byte oriented master and slave functions
- Full-duplex UART facilities
- Power control modes
  - Idle mode
  - Power-down mode
- Five package styles
- Extended temperature range
- OTP package available
- Three speed ranges
  - 16MHz
  - 20MHz

## LOGIC SYMBOL

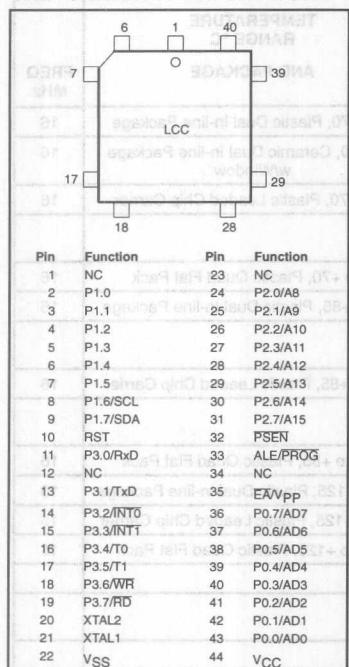


## PIN CONFIGURATIONS

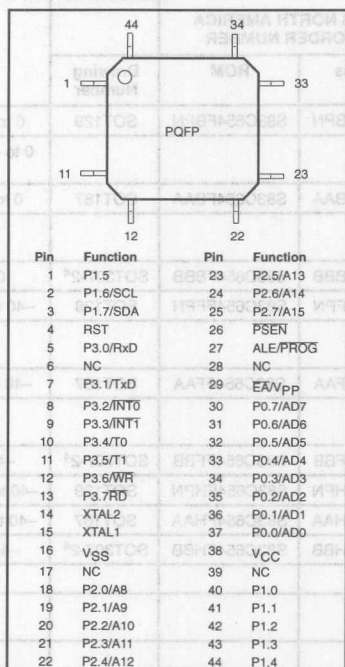


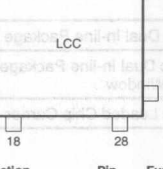
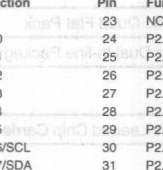
## CMOS single-chip 8-bit microcontroller 87C654

## CERAMIC AND PLASTIC LEADED CHIP CARRIER PIN FUNCTIONS



### PLASTIC QUAD FLAT PACK PIN FUNCTIONS



CHIP CARRIER PIN FUNCTIONS		PIN FUNCTIONS	
			
Pin	Function	Pin	Function
1	NC	23	NC
2	P1.0	24	P2.0/A8
3	P1.1	25	P2.1/A9
4	P1.2	26	P2.2/A10
5	P1.3	27	P2.3/A11
6	P1.4	28	P2.4/A12
7	P1.5	29	P2.5/A13
8	P1.6/SCL	30	P2.6/A14
9	P1.7/SDA	31	P2.7/A15
10	RST	32	PSEN
11	P3.0/RxD	33	ALE/PROG
12	NC	34	NC
13	P3.1/TxD	35	EA/V <sub>PP</sub>
14	P3.2/INT0	36	P0.7/AD7
15	P3.3/INT1	37	P0.6/AD6
16	P3.4/T0	38	P0.5/AD5
17	P3.5/T1	39	P0.4/AD4
18	P3.6/WR	40	P0.3/AD3
19	P3.7/RD	41	P0.2/AD2
20	XTAL2	42	P0.1/AD1
21	XTAL1	43	P0.0/AD0
22	VSS	44	VCC

## CMOS single-chip 8-bit microcontroller 87C654

## ORDERING INFORMATION

PHILIPS PART ORDER NUMBER PART MARKING		PHILIPS NORTH AMERICA PART ORDER NUMBER		Drawing Number	TEMPERATURE RANGE °C	FREQ MHz
ROMless	ROM	ROMless	ROM		AND PACKAGE	
P80C652FBP	P83C654FBP/xxx	S80C652FBPN	S83C654FBPN	SOT129	0 to +70, Plastic Dual In-line Package	16
					0 to +70, Ceramic Dual In-line Package w/Window	16
P80C652FBA	P83C654FBA/xxx	S80C652FBAA	S83C654FBAA	SOT187	0 to +70, Plastic Leaded Chip Carrier	16
P80C652FBB	P83C654FBB/xxx	S80C652FBBB	S83C654FBBB	SOT307-2 <sup>4</sup>	0 to +70, Plastic Quad Flat Pack	16
P80C652FFP	P83C654FFP/xxx	S80C652FFPN	S83C654FFPN	SOT129	-40 to +85, Plastic Dual In-line Package	16
P80C652FFA	P83C654FFA/xxx	S80C652FFAA	S83C654FFAA	SOT187	-40 to +85, Plastic Leaded Chip Carrier	16
P80C652FFB	P83C654FFB/xxx	S80C652FFBB	S83C654FFBB	SOT307-2 <sup>4</sup>	-40 to +85, Plastic Quad Flat Pack	16
P80C652FHP	P83C654FHP/xxx	S80C652FHPN	S83C654FHPN	SOT129	-40 to +125, Plastic Dual In-line Package	16
P80C652FHA	P83C654FHA/xxx	S80C652FHAA	S83C654FHAA	SOT187	-40 to +125, Plastic Leaded Chip Carrier	16
P80C652FHB	P83C654FHB/xxx	S80C652FHBB	S83C654FHBB	SOT307-2 <sup>4</sup>	-40 to +125, Plastic Quad Flat Pack	16
P80C652IBP	P83C654IBP/xxx	S80C652IBPN	S83C654IBPN	SOT129	0 to +70, Plastic Dual In-line Package	24
P80C652IBA	P83C654IBA/xxx	S80C652IBAA	S83C654IBAA	SOT187	0 to +70, Plastic Leaded Chip Carrier	24
P80C652IBB	P83C654IBB/xxx	S80C652IBBB	S83C654IBBB	SOT307-2 <sup>4</sup>	0 to +70, Plastic Quad Flat Pack	24
P80C652IFP	P83C654IFP/xxx	S80C652IFPN	S83C654IFPN	SOT129	-40 to +85, Plastic Dual In-line Package	24
P80C652IFA	P83C654IFA/xxx	S80C652IFAA	S83C654IFAA	SOT187	-40 to +85, Plastic Leaded Chip Carrier	24
P80C652IFB	P83C654IFB/xxx	S80C652IFBB	S83C654IFBB	SOT307-2 <sup>4</sup>	-40 to +85, Plastic Quad Flat Pack	24

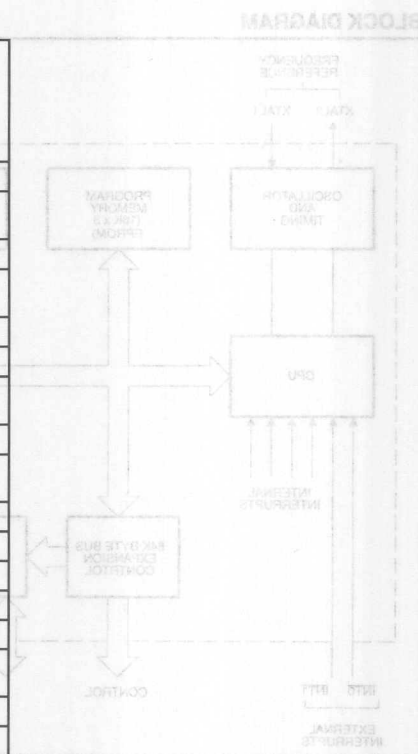
## NOTES:

- For full specification, see the 87C652 data sheet.
- 87C654 frequency range is 3.5MHz – 16MHz or 3.5MHz – 24MHz.
- xxx denotes the ROM code number.
- SOT311 replaced by SOT307-2.



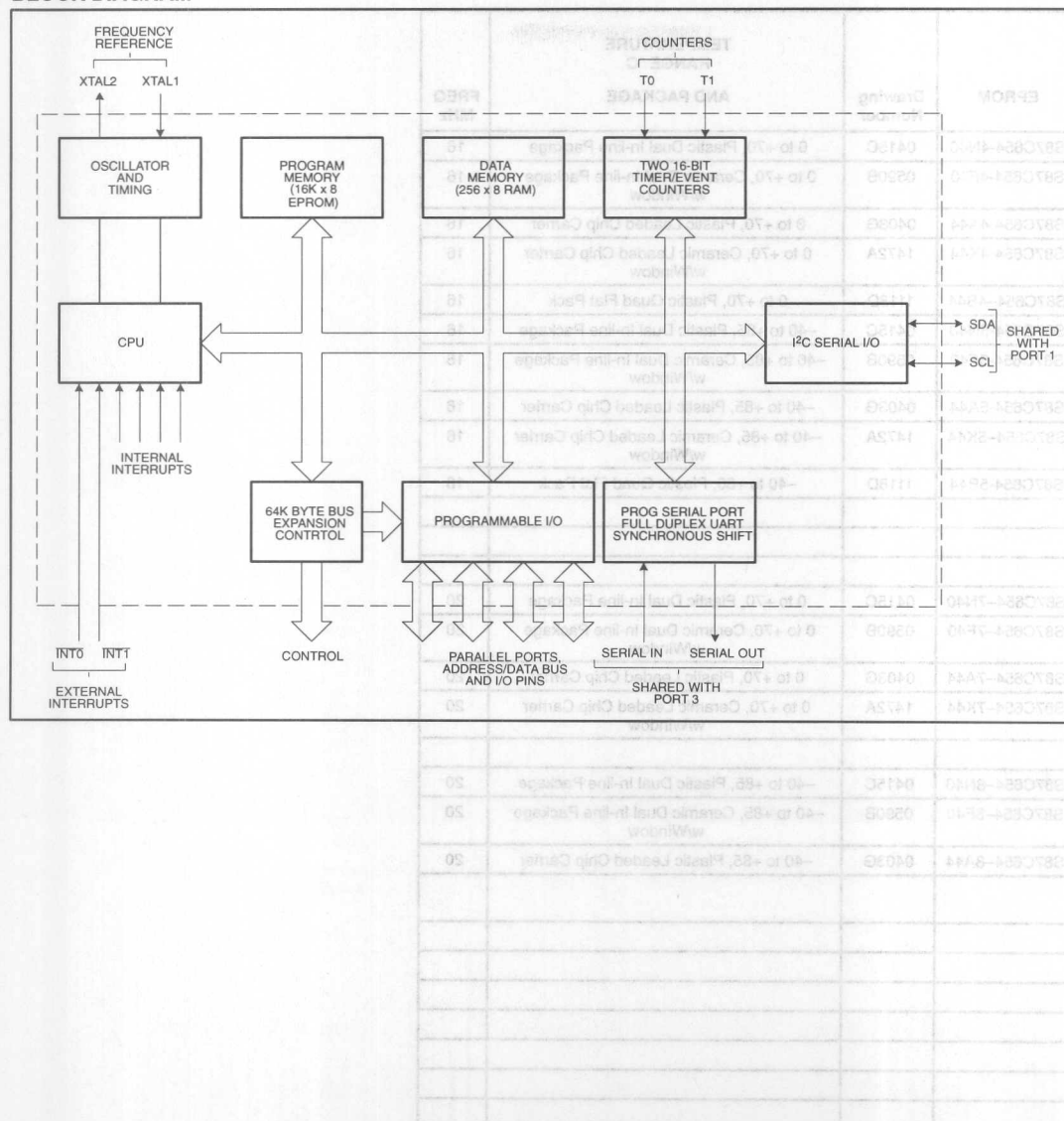
## CMOS single-chip 8-bit microcontroller S87C654

EPROM	Drawing Number	TEMPERATURE RANGE °C AND PACKAGE	FREQ MHz
S87C654-4N40	0415C	0 to +70, Plastic Dual In-line Package	16
S87C654-4F40	0590B	0 to +70, Ceramic Dual In-line Package w/Window	16
S87C654-4A44	0403G	0 to +70, Plastic Leaded Chip Carrier	16
S87C654-4K44	1472A	0 to +70, Ceramic Leaded Chip Carrier w/Window	16
S87C654-4B44	1118D	0 to +70, Plastic Quad Flat Pack	16
S87C654-5N40	0415C	-40 to +85, Plastic Dual In-line Package	16
S87C654-5F40	0590B	-40 to +85, Ceramic Dual In-line Package w/Window	16
S87C654-5A44	0403G	-40 to +85, Plastic Leaded Chip Carrier	16
S87C654-5K44	1472A	-40 to +85, Ceramic Leaded Chip Carrier w/Window	16
S87C654-5B44	1118D	-40 to +85, Plastic Quad Flat Pack	16
S87C654-7N40	0415C	0 to +70, Plastic Dual In-line Package	20
S87C654-7F40	0590B	0 to +70, Ceramic Dual In-line Package w/Window	20
S87C654-7A44	0403G	0 to +70, Plastic Leaded Chip Carrier	20
S87C654-7K44	1472A	0 to +70, Ceramic Leaded Chip Carrier w/Window	20
S87C654-8N40	0415C	-40 to +85, Plastic Dual In-line Package	20
S87C654-8F40	0590B	-40 to +85, Ceramic Dual In-line Package w/Window	20
S87C654-8A44	0403G	-40 to +85, Plastic Leaded Chip Carrier	20



## CMOS single-chip 8-bit microcontroller 87C654

### BLOCK DIAGRAM



## CMOS single-chip 8-bit microcontroller 87C654

## PIN DESCRIPTIONS

SYMBOL	DESCRIPTION	ADDRESS	DIRECT	BIT ADDRESS SYMBOL OR ALTERNATE	PIN NUMBER			TYPE	NAME AND FUNCTION
					DIP	LCC	QFP		
V <sub>SS</sub>					20	22	16	I	Ground: 0V reference.
V <sub>CC</sub>					40	44	38	I	<b>Power Supply:</b> This is the power supply voltage for normal, idle, and power-down operation.
P0.0–0.7					39–32	43–36	37–30	I/O	<b>Port 0:</b> Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s. Port 0 also outputs the code bytes during program verification in the 87C654. External pull-ups are required during program verification.
P1.0–P1.7					1–8	2–9	40–44, 1–3	I/O	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pull-ups, except P1.6 and P1.7 which are open drain. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 1 also receives the low-order address byte during program memory verification. Alternate functions include:
P1.6					7	8	2	I/O	<b>SCL:</b> I <sup>2</sup> C-bus serial port clock line.
P1.7					8	9	3	I/O	<b>SDA:</b> I <sup>2</sup> C-bus serial port data line.
P2.0–P2.7					21–28	24–31	18–25	I/O	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register.
P3.0–P3.7					10–17	11, 13–19	5, 7–13	I/O	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 3 also serves the special features of the 80C51 family, as listed below:
					10	11	5	I	<b>RxD (P3.0):</b> Serial input port
					11	13	7	O	<b>TxD (P3.1):</b> Serial output port
					12	14	8	I	<b>INT0 (P3.2):</b> External interrupt
					13	15	9	I	<b>INT1 (P3.3):</b> External interrupt
					14	16	10	I	<b>T0 (P3.4):</b> Timer 0 external input
					15	17	11	I	<b>T1 (P3.5):</b> Timer 1 external input
					16	18	12	O	<b>WR (P3.6):</b> External data memory write strobe
					17	19	13	O	<b>RD (P3.7):</b> External data memory read strobe
RST					9	10	4	I	<b>Reset:</b> A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to V <sub>SS</sub> permits a power-on reset using only an external capacitor to V <sub>CC</sub> .
ALE/PROG					30	33	27	I/O	<b>Address Latch Enable/Program Pulse:</b> Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory. This pin is also the program pulse input (PROG) during EPROM programming.
PSEN					29	32	26	O	<b>Program Store Enable:</b> The read strobe to external program memory. When the 87C654 is executing code from the external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory. PSEN is not activated during fetches from internal program memory.
EA/V <sub>PP</sub>					31	35	29	I	<b>External Access Enable/Programming Supply Voltage:</b> EA must be externally held low to enable the device to fetch code from external program memory locations 0000H and 3FFFH. If EA is held high, the device executes from internal program memory unless the program counter contains an address greater than 3FFFH. This pin also receives the 12.75V programming supply voltage (V <sub>PP</sub> ) during EPROM programming.
XTAL1					19	21	15	I	<b>Crystal 1:</b> Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2					18	20	14	O	<b>Crystal 2:</b> Output from the inverting oscillator amplifier.

## NOTE:

To avoid "latch-up" effect at power-on, the voltage on any pin at any time must not be higher than V<sub>CC</sub> + 0.5V or V<sub>SS</sub> – 0.5V, respectively.

## CMOS single-chip 8-bit microcontroller 87C654

Table 1. 87C652/654 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB				LSB				
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
DPTR:	Data pointer (2 bytes)										
DPH	Data pointer high	83H									00H
DPL	Data pointer low	82H									00H
			AF	AE	AD	AC	AB	AA	A9	A8	
IE*#	Interrupt enable	A8H	EA		ES1	ES0	ET1	EX1	ET0	EX0	0x000000B
			BF	BE	BD	BC	BB	BA	B9	B8	
IP*#	Interrupt priority	B8H	—		PS1	PS0	PT1	PX1	PT0	PX0	xx000000B
			87	86	85	84	83	82	81	80	
P0*	Port 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
			97	96	95	94	93	92	91	90	
P1*#	Port 1	90H	SDA	SCL							FFH
			A7	A6	A5	A4	A3	A2	A1	A0	
P2*	Port 2	A0H	A15	A14	A13	A12	A11	A10	A9	A8	FFH
			B7	B6	B5	B4	B3	B2	B1	B0	
P3*	Port 3	B0H	RD	WR	T1	T0	INT1	INT0	TXD	RXD	FFH
PCON	Power control	87H	SMOD	—	—	—	GF1	GF0	PD	IDL	0xxx0000B
			9F	9E	9D	9C	9B	9A	99	98	
SOCON*#	Serial 0 port control	98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00H
S0BUF#	Serial 0 data buffer	99H									xxxxxxx8B
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program status word	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	00H
S1DAT#	Serial 1 data	DAH									00H
SP	Stack pointer	81H									07H
S1ADR#	Serial 1 address	DBH	SLAVE ADDRESS							GC	00H
S1STA#	Serial 1 status	D9H	SC4	SC3	SC2	SC1	SC0	0	0	0	F8H
			DF	DE	DD	DC	DB	DA	D9	D8	
S1CON*#	Serial 1 control	D8H	CR2	ENS1	STA	STO	SI	AA	CR1	CR0	00000000B
			8F	8E	8D	8C	8B	8A	89	88	
TCON*	Timer control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
TH1	Timer high 1	8DH									00H
TH0	Timer high 0	8CH									00H
TL1	Timer low 1	8BH									00H
TH0	Timer low 0	8AH									00H
TMOD	Timer mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.

## CMOS single-chip 8-bit microcontroller 87C654

**OSCILLATOR CHARACTERISTICS**

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator, as shown in the Logic Symbol.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

**Reset**

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. To insure a good power-on reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few

milliseconds) plus two machine cycles. At power-on, the voltage on V<sub>CC</sub> and RST must come up at the same time for a proper start-up.

**Idle Mode**

In the idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

**Power-Down Mode**

In the power-down mode, the oscillator is stopped and the instruction to invoke

power-down is the last instruction executed. Only the contents of the on-chip RAM are preserved. A hardware reset is the only way to terminate the power-down mode. The control bits for the reduced power modes are in the special function register PCON. Table 2 shows the state of the I/O ports during low current operating modes.

**I<sup>2</sup>C SERIAL COMMUNICATION—SIO1**

The I<sup>2</sup>C serial port is identical to the I<sup>2</sup>C serial port on the 8XC552. The operation of this subsystem is described in detail in the 8XC552 section of this manual.

Note that in both the 8XC652/4 and the 8XC552 the I<sup>2</sup>C pins are alternate functions to port pins P1.6 and P1.7. Because of this, P1.6 and P1.7 on these parts do not have a pull-up structure as found on the 80C51. Therefore P1.6 and P1.7 have open drain outputs on the 8XC652/4.

**Table 2. External Pin Status During Idle and Power-Down Mode**

MODE	PROGRAM MEMORY	TEMPERATURE RANGE		FREQUENCY (MHz)		SUPPLY VOLTAGE (V)		TYPE
		MIN	MAX	MIN	MAX	MIN	MAX	
Idle	Internal	1	1	Data	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data	Data
Power-down	Internal	0	0	Data	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data	Data

**Serial Control Register (S1CON) – See Table 3**

S1CON (D8H)	CR2	ENS1	STA	STO	SI	AA	CR1	CR0
-------------	-----	------	-----	-----	----	----	-----	-----

Bits CR0, CR1 and CR2 determine the serial clock frequency that is generated in the master mode of operation.

**Table 3. Serial Clock Rates**

CR2	CR1	CR0	BIT FREQUENCY (kHz) AT f <sub>OSC</sub>				f <sub>OSC</sub> DIVIDED BY
			6MHz	12MHz	16MHz	20MHz	
0	0	0	23	47	62.5	78	256
0	0	1	27	54	71	89 <sup>1</sup>	224
0	1	0	31.25	62.5	83.3	104 <sup>1</sup>	192
0	1	1	37	75	100	125 <sup>1</sup>	160
1	0	0	6.25	12.5	17	21	960
1	0	1	50	100	133 <sup>1</sup>	166 <sup>1</sup>	120
1	1	0	100	200 <sup>1</sup>	267 <sup>1</sup>	334 <sup>1</sup>	60
1	1	1	0.25 < 62.5 0 to 255	0.5 < 62.5 0 to 254	0.65 < 55.6 0 to 253	0.81 < 69.4 0 to 253	96 × (256 – (reload value Timer 1)) (Reload value range: 0 – 254 in mode 2)

**NOTES:**

1. These frequencies exceed the upper limit of 100kHz of the I<sup>2</sup>C-bus specification and cannot be used in an I<sup>2</sup>C-bus application.



## CMOS single-chip 8-bit microcontroller 87C654

**ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>**

PARAMETER	RATING	UNIT
Storage temperature range	-65 to +150	°C
Voltage on $\overline{EAV}_{PP}$ to $V_{SS}$	-0.5 to + 13	V
Voltage on any other pin to $V_{SS}$	-0.5 to + 6.5	V
Input, output current on any single pin	±5	mA
Power dissipation (based on package heat transfer limitations, not device power consumption)	1	W

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
2. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
3. Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to  $V_{SS}$  unless otherwise noted.

## DEVICE SPECIFICATIONS

TYPE	SUPPLY VOLTAGE (V)		FREQUENCY (MHz)		TEMPERATURE RANGE (°C)
	MIN.	MAX.	MIN.	MAX.	
S87C654-4	4.5	5.5	3.5	16	0 to +70
S87C654-5	4.5	5.5	3.5	16	-40 to +85
S87C654-7	4.5	5.5	3.5	20	0 to +70
S87C654-8	4.5	5.5	3.5	20	-40 to +85

## CMOS single-chip 8-bit microcontroller 87C654

## DC ELECTRICAL CHARACTERISTICS

V<sub>SS</sub> = 0V

SYMBOL	PARAMETER	PART TYPE	TEST CONDITIONS	LIMITS		UNIT
				MIN.	MAX.	
V <sub>IL</sub>	Input low voltage, except EA, P1.6/SCL, P1.7/SDA	0 to +70°C -40 to +85°C		-0.5 -0.5	0.2V <sub>CC</sub> -0.1 0.2V <sub>CC</sub> -0.15	V V
V <sub>IL1</sub>	Input low voltage to EA	0 to +70°C -40 to +85°C		-0.5 -0.5	0.2V <sub>CC</sub> -0.3 0.2V <sub>CC</sub> -0.35	V V
V <sub>IL2</sub>	Input low voltage to P1.6/SCL, P1.7/SDA <sup>1</sup>			-0.5	0.3V <sub>CC</sub>	V
V <sub>IH</sub>	Input high voltage, except XTAL1, RST, P1.6/SCL, P1.7/SDA	0 to +70°C -40 to +85°C		0.2V <sub>CC</sub> +0.9 0.2V <sub>CC</sub> +1.0	V <sub>CC</sub> +0.5 V <sub>CC</sub> +0.5	V V
V <sub>IH1</sub>	Input high voltage, XTAL1, RST	0 to +70°C -40 to +85°C		0.7V <sub>CC</sub> 0.7V <sub>CC</sub> +0.1	V <sub>CC</sub> +0.5 V <sub>CC</sub> +0.5	V V
V <sub>IH2</sub>	Input high voltage, P1.6/SCL, P1.7/SDA <sup>1</sup>			0.7V <sub>CC</sub>	6.0	V
V <sub>OL</sub>	Output low voltage, ports 1, 2, 3, except P1.6/SCL, P1.7/SDA		I <sub>OL</sub> = 1.6mA <sup>2, 3</sup>		0.45	V
V <sub>OL1</sub>	Output low voltage, port 0, ALE, PSEN		I <sub>OL</sub> = 3.2mA <sup>2, 3</sup>		0.45	V
V <sub>OL2</sub>	Output low voltage, P1.6/SCL, P1.7/SDA		I <sub>OL</sub> = 3.0mA		0.4	V
V <sub>OH</sub>	Output high voltage, ports 1, 2, 3	0 to +70°C -40 to +85°C	I <sub>OH</sub> = -60µA I <sub>OH</sub> = -25µA	2.4 0.75V <sub>CC</sub>		V V
V <sub>OH1</sub>	Output high voltage; port 0 in external bus mode, ALE, PSEN, RST <sup>4</sup>	0 to +70°C -40 to +85°C	I <sub>OH</sub> = -400µA I <sub>OH</sub> = -150µA	2.4 0.75V <sub>CC</sub>		V V
I <sub>IL</sub>	Logical 0 input current, ports 1, 2, 3, 4, except P1.6/SCL, P1.7/SDA	0 to +70°C -40 to +85°C	V <sub>IN</sub> = 0.45V		-50 -75	µA µA
I <sub>TL</sub>	Logical 1-to-0 transition current, ports 1, 2, 3, except P1.6/SCL, P1.7/SDA	0 to +70°C -40 to +85°C	See note 5		-650 -750	µA µA
I <sub>L1</sub>	Input leakage current, port 0		0.45V < V <sub>I</sub> < V <sub>CC</sub>		±10	µA
I <sub>L2</sub>	Input leakage current, P1.6/SCL, P1.7/SDA		0V < V <sub>I</sub> < 6.0V 0V < V <sub>CC</sub> < 6.0V		±10	µA µA
I <sub>CC</sub>	Power supply current:  Active mode @ 16MHz <sup>7</sup> Idle mode @ 16MHz <sup>8</sup> Power down mode <sup>9, 10</sup> Power down mode <sup>9, 10</sup>	0 to +70°C -40 to +85°C	See note 6 V <sub>CC</sub> =6.0V		25 6 50 135	mA mA µA µA
R <sub>RST</sub>	Internal reset pull-down resistor			50	150	kΩ
C <sub>IO</sub>	Pin capacitance		Freq.=1MHz		10	pF

NOTES: See Next Page.

## CMOS single-chip 8-bit microcontroller 87C654

## NOTES FOR DC ELECTRICAL CHARACTERISTICS:

- The input threshold voltage of P1.6 and P1.7 (SI01) meets the I<sup>2</sup>C specification, so an input voltage below 0.3V<sub>CC</sub> will be recognized as a logic 0 while an input voltage above 0.7V<sub>CC</sub> will be recognized as a logic 1.
- Capacitive loading on ports 0 and 2 may cause spurious noise to be superimposed on the V<sub>OL</sub>s of ALE and ports 1 and 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100pF), the noise pulse on the ALE pin may exceed 0.8V. In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input. I<sub>OL</sub> can exceed these conditions provided that no single output sinks more than 5mA and no more than two outputs exceed the test conditions.
- Under steady state (non-transient) conditions, I<sub>OL</sub> must be externally limited as follows: Maximum I<sub>OL</sub> = 10mA per port pin; Maximum I<sub>OL</sub> = 26mA total for Port 0; Maximum I<sub>OL</sub> = 15mA total for Ports 1, 2, and 3; Maximum I<sub>OL</sub> = 71mA total for all output pins. If I<sub>OL</sub> exceeds the test conditions, V<sub>OL</sub> may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
- Capacitive loading on ports 0 and 2 may cause the V<sub>OH</sub> on ALE and PSEN to momentarily fall below the 0.9V<sub>CC</sub> specification when the address bits are stabilizing.
- Pins of ports 1, 2, and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when V<sub>IN</sub> is approximately 2V.
- See Figures 9 through 11 for I<sub>CC</sub> test conditions.
- The operating supply current is measured with all output pins disconnected; XTAL1 driven with t<sub>r</sub> = t<sub>f</sub> = 10ns; V<sub>IL</sub> = V<sub>SS</sub> + 0.5V; V<sub>IH</sub> = V<sub>CC</sub> - 0.5V; XTAL2 not connected; EA = RST = Port 0 = P1.6 = P1.7 = V<sub>CC</sub>; f<sub>CLK</sub> = 16MHz. See Figure 9.
- The idle mode supply current is measured with all output pins disconnected; XTAL1 driven with t<sub>r</sub> = t<sub>f</sub> = 10ns; V<sub>IL</sub> = V<sub>SS</sub> + 0.5V; V<sub>IH</sub> = V<sub>CC</sub> - 0.5V; XTAL2 not connected; Port 0 = P1.6 = P1.7 = V<sub>CC</sub>; EA = RST = V<sub>SS</sub>; f<sub>CLK</sub> = 16MHz. See Figure 10.
- The power-down current is measured with all output pins disconnected; XTAL2 not connected; Port 0 = P1.6 = P1.7 = V<sub>CC</sub>; EA = RST = V<sub>SS</sub>. See Figure 11.
- 2V ≤ V<sub>PD</sub> ≤ V<sub>CCmax</sub>.

Symbol	Parameter	Conditions	Min	Typ	Max
I <sub>CC</sub>	Power supply current	Active mode @ 16MHz Idle mode @ 16MHz Power-down mode, 10 Power-down mode, 10		0.10 to 0.70 -40 to +85°C	
I <sub>CS</sub>	Input leakage current, P1.6/CSCL, P1.7/SDA		0V < V <sub>CC</sub> < 8.0V 0V < V <sub>CC</sub> < 8.0V	±10	
I <sub>IL</sub>	Input leakage current, port 0		0.45V < V <sub>I</sub> < V <sub>CC</sub>	±10	
I <sub>IL</sub>	Logical 1-to-0 transition current, ports 1, 2, 3, 4	except P1.6/CSCL, P1.7/SDA	0 to +70°C	-800 to -500	
I <sub>IL</sub>	Logical 0 input current, ports 1, 2, 3, 4	except P1.6/CSCL, P1.7/SDA	0 to +70°C	-80 to -50	
V <sub>OH</sub>	Output high voltage, ports 1, 2, 3		0 to +70°C	0.75V <sub>CC</sub>	5.4
V <sub>OH</sub>	Output high voltage, port 0 in external bus mode		0 to +70°C	0.75V <sub>CC</sub>	5.4
V <sub>OL</sub>	Output low voltage, P1.6/CSCL, P1.7/SDA		0 to +70°C	0.4	
V <sub>OL</sub>	Output low voltage, port 0, ALE, PSEN		0 to +70°C	0.4	
I <sub>OL</sub>	Output low current, P1.6/CSCL, P1.7/SDA		0 to +70°C	3.0mA	
I <sub>OL</sub>	Output low current, port 0, ALE, PSEN		0 to +70°C	3.0mA	
f <sub>osc</sub>	Frequency	Fixed = 16MHz		10	

NOTES: See next page.

## CMOS single-chip 8-bit microcontroller 87C654

AC ELECTRICAL CHARACTERISTICS<sup>1, 2</sup>

SYMBOL	FIGURE	PARAMETER	16MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
$f_{\text{CLCL}}$	2	Oscillator frequency Speed Versions 87C654 -4, -5			3.5	16	MHz
$t_{\text{HLL}}$	2	ALE pulse width	85		$2t_{\text{CLCL}}-40$		ns
$t_{\text{AVLL}}$	2	Address valid to ALE low	8		$t_{\text{CLCL}}-55$		ns
$t_{\text{LLAX}}$	2	Address hold after ALE low	28		$t_{\text{CLCL}}-35$		ns
$t_{\text{LLIV}}$	2	ALE low to valid instruction in		150		$4t_{\text{CLCL}}-100$	ns
$t_{\text{LLPL}}$	2	ALE low to PSEN low	23		$t_{\text{CLCL}}-40$		ns
$t_{\text{PLPH}}$	2	PSEN pulse width	143		$3t_{\text{CLCL}}-45$		ns
$t_{\text{PLIV}}$	2	PSEN low to valid instruction in		83		$3t_{\text{CLCL}}-105$	ns
$t_{\text{PXIX}}$	2	Input instruction hold after PSEN	0				ns
$t_{\text{PXIZ}}$	2	Input instruction float after PSEN		38		$t_{\text{CLCL}}-25$	ns
$t_{\text{AVIV}}$	2	Address to valid instruction in		208		$5t_{\text{CLCL}}-105$	ns
$t_{\text{PLAZ}}$	2	PSEN low to address float		10		10	ns
<b>Data Memory</b>							
$t_{\text{AVLL}}$	3, 4	Address valid to ALE low	28		$t_{\text{CLCL}}-35$		ns
$t_{\text{RLRH}}$	3, 4	RD pulse width	275		$6t_{\text{CLCL}}-100$		ns
$t_{\text{WLWH}}$	3, 4	WR pulse width	275		$6t_{\text{CLCL}}-100$		ns
$t_{\text{RLDV}}$	3, 4	RD low to valid data in		148		$5t_{\text{CLCL}}-165$	ns
$t_{\text{RHDZ}}$	3, 4	Data hold after RD	0		0		ns
$t_{\text{RHDZ}}$	3, 4	Data float after RD		55		$2t_{\text{CLCL}}-70$	ns
$t_{\text{LLDV}}$	3, 4	ALE low to valid data in		350		$8t_{\text{CLCL}}-150$	ns
$t_{\text{AVDV}}$	3, 4	Address to valid data in		398		$9t_{\text{CLCL}}-165$	ns
$t_{\text{LLWL}}$	3, 4	ALE low to RD or WR low	138	238	$3t_{\text{CLCL}}-50$	$3t_{\text{CLCL}}+50$	ns
$t_{\text{AVWL}}$	3, 4	Address valid to WR low or RD low	120		$4t_{\text{CLCL}}-130$		ns
$t_{\text{QVWX}}$	3, 4	Data valid to WR transition	3		$t_{\text{CLCL}}-60$		ns
$t_{\text{DW}}$	3, 4	Data setup time before WR	288		$7t_{\text{CLCL}}-150$		ns
$t_{\text{WHQX}}$	3, 4	Data hold after WR	13		$t_{\text{CLCL}}-50$		ns
$t_{\text{RLAZ}}$	3, 4	RD low to address float		0		0	ns
$t_{\text{WHLH}}$	3, 4	RD or WR high to ALE high	23	103	$t_{\text{CLCL}}-40$	$t_{\text{CLCL}}+40$	ns
<b>Shift Register</b>							
$t_{\text{XLXL}}$	5	Serial port clock cycle time <sup>3</sup>	0.75		$12t_{\text{CLCL}}$		μs
$t_{\text{QVXH}}$	5	Output data setup to clock rising edge <sup>3</sup>	492		$10t_{\text{CLCL}}-133$		ns
$t_{\text{XHDX}}$	5	Output data hold after clock rising edge <sup>3</sup>	80		$2t_{\text{CLCL}}-117$		ns
$t_{\text{XHDX}}$	5	Input data hold after clock rising edge <sup>3</sup>	0		0		ns
$t_{\text{XHDV}}$	5	Clock rising edge to input data valid <sup>3</sup>		492		$10t_{\text{CLCL}}-133$	ns
<b>External Clock</b>							
$t_{\text{CHCX}}$	6	High time <sup>3</sup>	20		20	$t_{\text{CLCL}} - t_{\text{LOW}}$	ns
$t_{\text{CLCX}}$	6	Low time <sup>3</sup>	20		20	$t_{\text{CLCL}} - t_{\text{HIGH}}$	ns
$t_{\text{CLCH}}$	6	Rise time <sup>3</sup>		20		20	ns
$t_{\text{CHCL}}$	6	Fall time <sup>3</sup>		20		20	ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- These values are characterized but not 100% production tested.

## CMOS single-chip 8-bit microcontroller 87C654

AC ELECTRICAL CHARACTERISTICS<sup>1, 2</sup>

SYMBOL	FIGURE	PARAMETER	20MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
$1/t_{CLCL}$	2	Oscillator frequency: Speed Versions 87C654 -7, -8			3.5	20	MHz
$t_{LHLL}$	2	ALE pulse width	60		$2t_{CLCL}-40$		ns
$t_{AVLL}$	2	Address valid to ALE low	25		$t_{CLCL}-25$		ns
$t_{LLAX}$	2	Address hold after ALE low	25		$t_{CLCL}-25$		ns
$t_{LLIV}$	2	ALE low to valid instruction in		135		$4t_{CLCL}-65$	ns
$t_{LLPL}$	2	ALE low to PSEN low	25		$t_{CLCL}-25$		ns
$t_{PLPH}$	2	PSEN pulse width	105		$3t_{CLCL}-45$		ns
$t_{PLIV}$	2	PSEN low to valid instruction in		90		$3t_{CLCL}-60$	ns
$t_{PXIX}$	2	Input instruction hold after PSEN		0			ns
$t_{PXIZ}$	2	Input instruction float after PSEN		25		$t_{CLCL}-25$	ns
$t_{AVIV}$	2	Address to valid instruction in		170		$5t_{CLCL}-80$	ns
$t_{PLAZ}$	2	PSEN low to address float		10		10	ns
Data Memory							
$t_{AVLL}$	3, 4	Address valid to ALE low	25		$t_{CLCL}-25$		ns
$t_{RLRH}$	3, 4	RD pulse width	200		$6t_{CLCL}-100$		ns
$t_{WLWH}$	3, 4	WR pulse width	200		$6t_{CLCL}-100$		ns
$t_{RLDV}$	3, 4	RD low to valid data in		160		$5t_{CLCL}-90$	ns
$t_{RHDX}$	3, 4	Data hold after RD	0		0		ns
$t_{RHDX}$	3, 4	Data float after RD		72		$2t_{CLCL}-28$	ns
$t_{LLDV}$	3, 4	ALE low to valid data in		250		$8t_{CLCL}-150$	ns
$t_{AVDV}$	3, 4	Address to valid data in		285		$9t_{CLCL}-165$	ns
$t_{LLWL}$	3, 4	ALE low to RD or WR low	100	200	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
$t_{AVWL}$	3, 4	Address valid to WR low or RD low	125		$4t_{CLCL}-75$		ns
$t_{QVWX}$	3, 4	Data valid to WR transition	20		$t_{CLCL}-30$		ns
$t_{DW}$	3, 4	Data setup time before WR	220		$7t_{CLCL}-130$		ns
$t_{WHQX}$	3, 4	Data hold after WR	25		$t_{CLCL}-25$		ns
$t_{RLAZ}$	3, 4	RD low to address float		0		0	ns
$t_{WHLH}$	3, 4	RD or WR high to ALE high	25	75	$t_{CLCL}-25$	$t_{CLCL}+25$	ns
Shift Register							
$t_{XLXL}$	5	Serial port clock cycle time <sup>3</sup>	0.6		$12t_{CLCL}$		µs
$t_{QVXH}$	5	Output data setup to clock rising edge <sup>3</sup>	367		$10t_{CLCL}-133$		ns
$t_{XHDX}$	5	Output data hold after clock rising edge <sup>3</sup>	40		$2t_{CLCL}-60$		ns
$t_{XHDV}$	5	Input data hold after clock rising edge <sup>3</sup>	0		0		ns
$t_{XHDV}$	5	Clock rising edge to input data valid <sup>3</sup>		367		$10t_{CLCL}-133$	ns
External Clock							
$t_{CHCX}$	6	High time <sup>3</sup>	17		17	$t_{CLCL}-t_{LOW}$	ns
$t_{CLCX}$	6	Low time <sup>3</sup>	17		17	$t_{CLCL}-t_{HIGH}$	ns
$t_{CLCH}$	6	Rise time <sup>3</sup>		20		20	ns
$t_{CHCL}$	6	Fall time <sup>3</sup>		20		20	ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- These values are characterized but not 100% production tested.



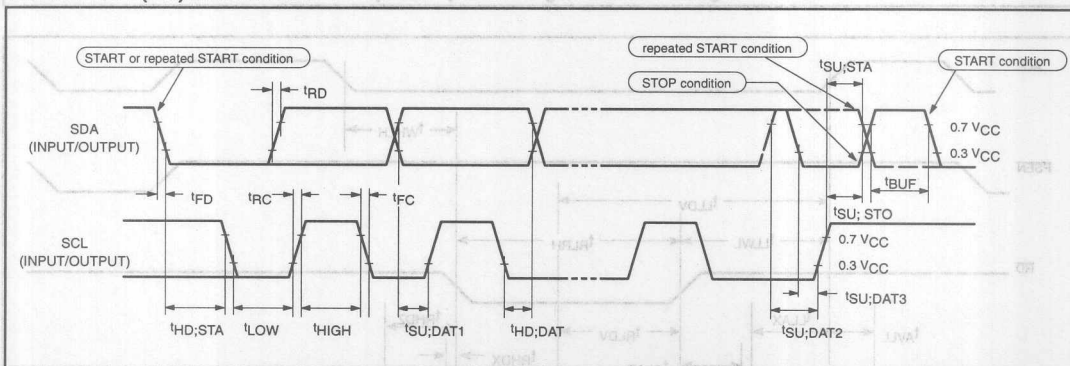
## CMOS single-chip 8-bit microcontroller 87C654

AC ELECTRICAL CHARACTERISTICS – I<sup>2</sup>C INTERFACE

SYMBOL	PARAMETER	INPUT	OUTPUT
<b>SCL TIMING CHARACTERISTICS</b>			
$t_{HD; STA}$	START condition hold time	$\geq 14 t_{CLCL}$	$> 4.0\mu s^1$
$t_{LOW}$	SCL LOW time	$\geq 16 t_{CLCL}$	$> 4.7\mu s^1$
$t_{HIGH}$	SCL HIGH time	$\geq 14 t_{CLCL}$	$> 4.0\mu s^1$
$t_{RC}$	SCL rise time	$\leq 1\mu s$	$-^2$
$t_{FC}$	SCL fall time	$\leq 0.3\mu s$	$< 0.3\mu s^3$
<b>SDA TIMING CHARACTERISTICS</b>			
$t_{SU; DAT1}$	Data set-up time	$\geq 250ns$	$> 20 t_{CLCL} - t_{RD}$
$t_{SU; DAT2}$	SDA set-up time (before rep. START cond.)	$\geq 250ns$	$> 1\mu s^1$
$t_{SU; DAT3}$	SDA set-up time (before STOP cond.)	$\geq 250ns$	$> 8 t_{CLCL}$
$t_{HD; DAT}$	Data hold time	$\geq 0ns$	$> 8 t_{CLCL} - t_{FC}$
$t_{SU; STA}$	Repeated START set-up time	$\geq 14 t_{CLCL}$	$> 4.7\mu s^1$
$t_{SU; STO}$	STOP condition set-up time	$\geq 14 t_{CLCL}$	$> 4.0\mu s^1$
$t_{BUF}$	Bus free time	$\geq 14 t_{CLCL}$	$> 4.7\mu s^1$
$t_{RD}$	SDA rise time	$\leq 1\mu s$	$-^2$
$t_{FD}$	SDA fall time	$\leq 0.3\mu s$	$< 0.3\mu s^3$

## NOTES:

- At 100 kbit/s. At other bit rates this value is inversely proportional to the bit-rate of 100 kbit/s.
- Determined by the external bus-line capacitance and the external bus-line pull-resistor, this must be  $< 1\mu s$ .
- Spikes on the SDA and SCL lines with a duration of less than  $3 t_{CLCL}$  will be filtered out. Maximum capacitance on bus-lines SDA and SCL = 400pF.
- $t_{CLCL} = 1/f_{OSC}$  = one oscillator clock period at pin XTAL1. For  $62ns < t_{CLCL} < 285ns$  (16MHz)  $> f_{OSC} > 3.5MHz$  the SI01 interface meets the I<sup>2</sup>C-bus specification for bit-rates up to 100 kbit/s.

TIMING SI01 (I<sup>2</sup>C) INTERFACE

## CMOS single-chip 8-bit microcontroller 87C654

## EXPLANATION OF THE AC SYMBOLS

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

A - Address

C - Clock

D – Input data

H – Logic level high

- 1 - Instruction (program memory contents)

L – Logic level low, or ALE

P - PSEN

Q – Output data

R –  $\overline{RD}$  signal

 $t - T_1$ 

Y – Valid

W -  $\overline{WR}$  signal

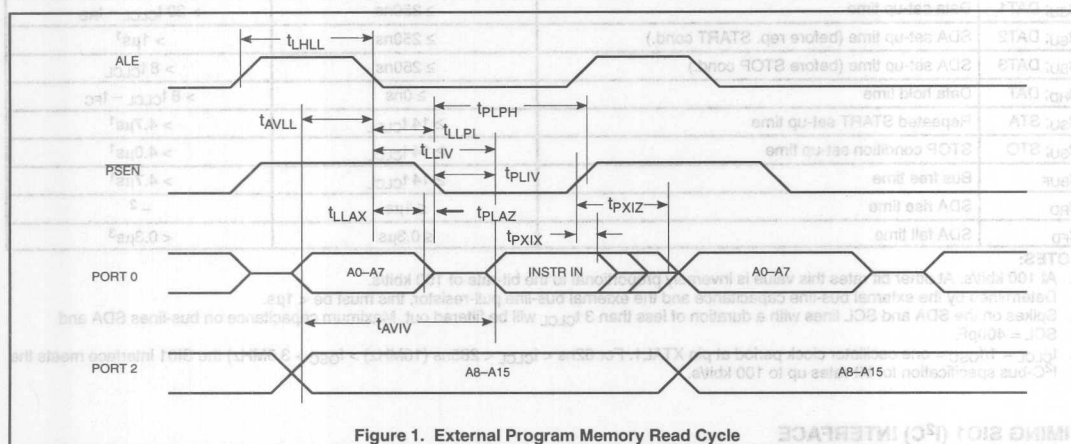
X – No longer a valid logic level

Z - Float

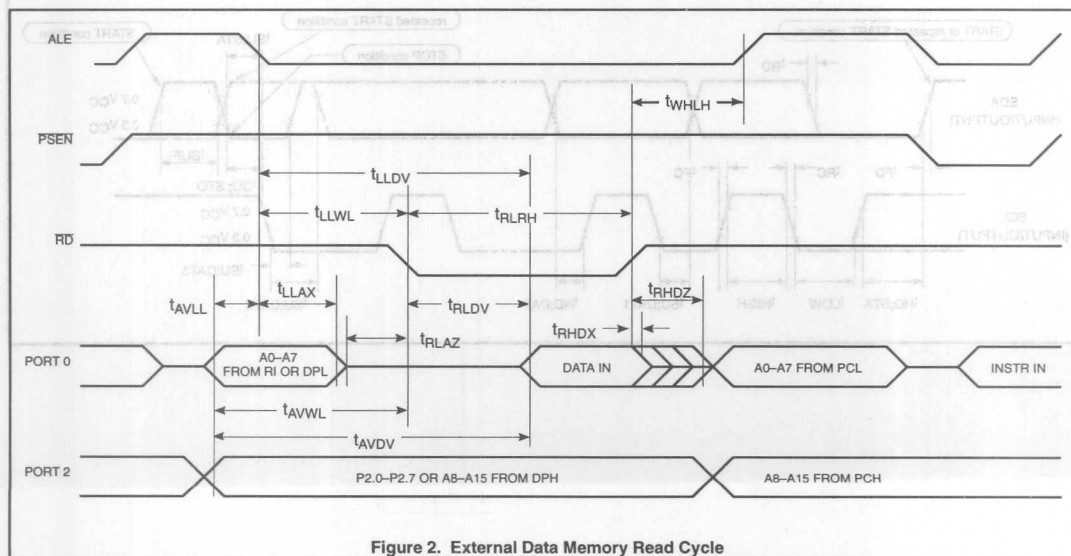
**Examples:**  $t_{AVLL}$  = Time for address valid to ALE low.

 $t_{l \mid p_l}$  = Time for ALE low

to  $\overline{\text{PSEN}}$  low.

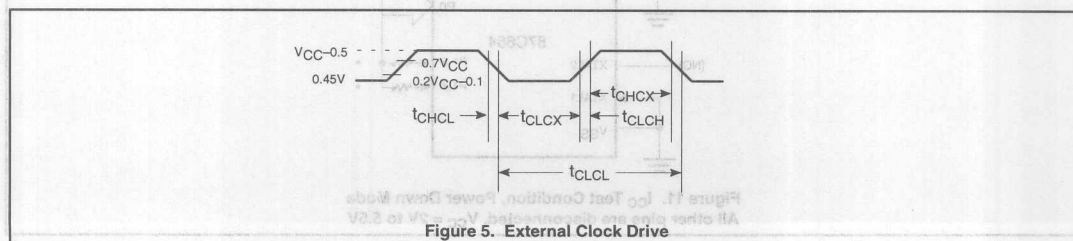
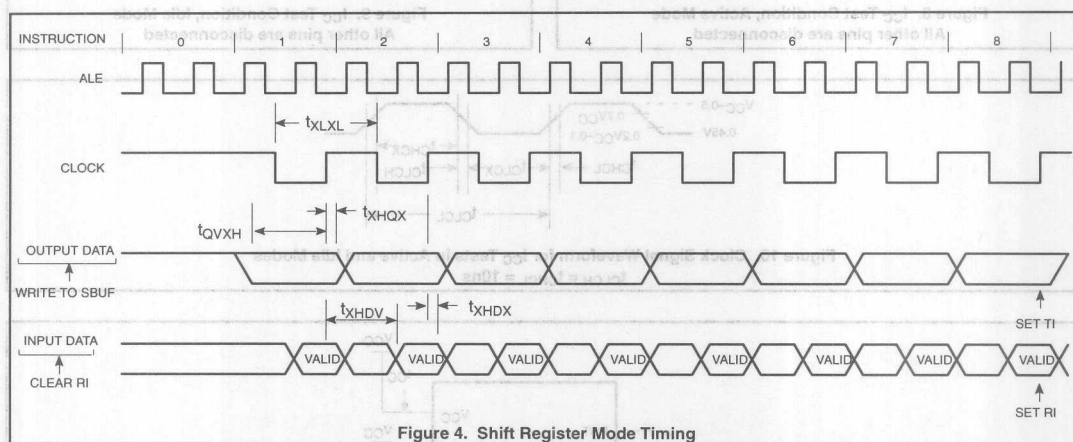
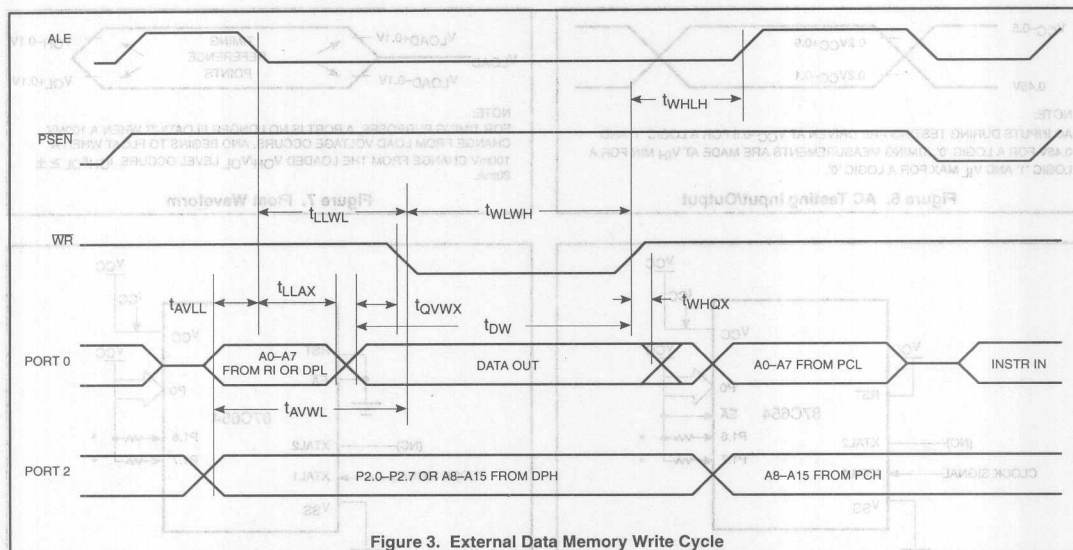


**Figure 1. External Program Memory Read Cycle**



### Figure 2. External Data Memory Read Cycle

## CMOS single-chip 8-bit microcontroller 87C654



## CMOS single-chip 8-bit microcontroller 87C654

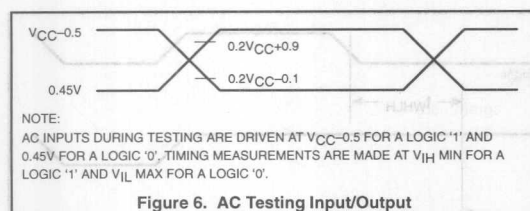


Figure 6. AC Testing Input/Output

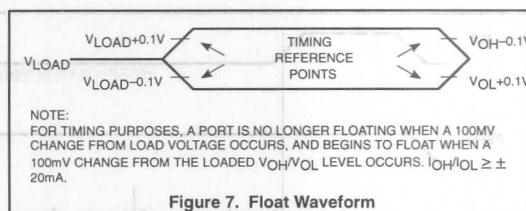
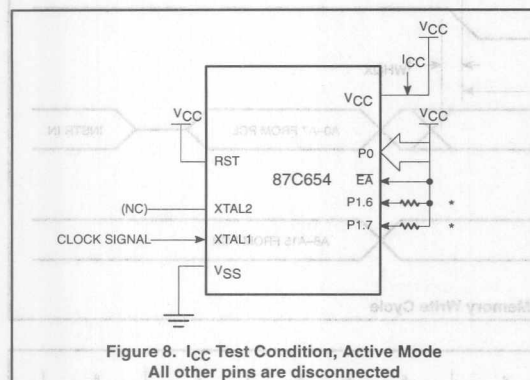
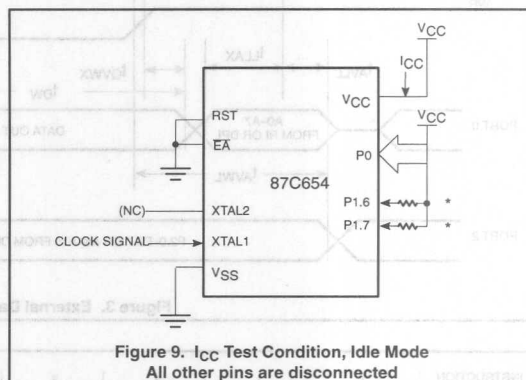


Figure 7. Float Waveform



**Figure 8. ICC Test Condition, Active Mode**  
All other pins are disconnected



**Figure 9. ICC Test Condition, Idle Mode**  
All other pins are disconnected

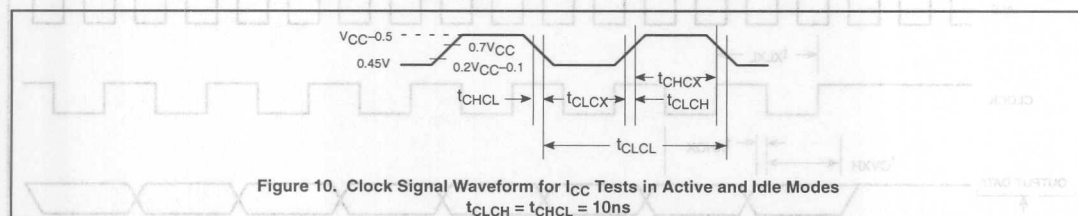
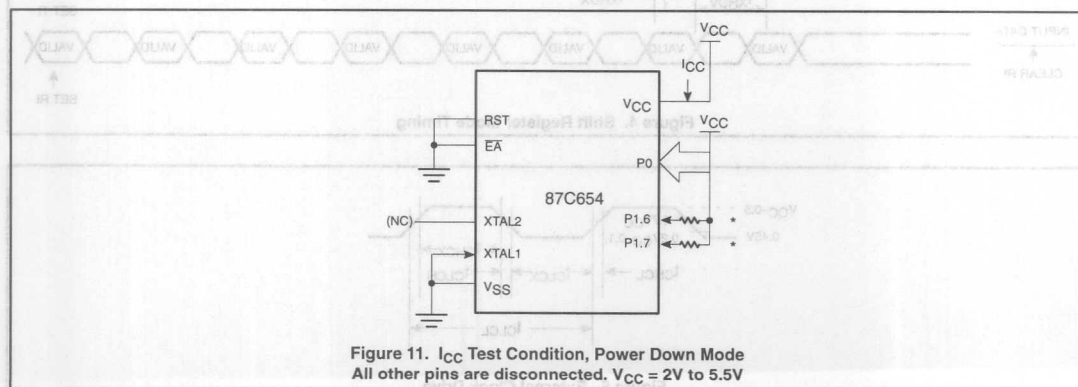


Figure 10. Clock Signal Waveform for I<sub>CC</sub> Tests in Active and Idle Modes  
 $t_{CLCH} = t_{CHCL} = 10\text{ns}$



**Figure 11. I<sub>CC</sub> Test Condition, Power Down Mode**  
All other pins are disconnected. V<sub>CC</sub> = 2V to 5.5V

**NOTE:**

\* Ports 1.6 and 1.7 should be connected to  $V_{CC}$  through resistors of sufficiently high value such that the sink current into these pins does not exceed the  $I_{OL1}$  specification.

## CMOS single-chip 8-bit microcontroller 87C654

### EPROM CHARACTERISTICS

The 87C654 is programmed by using a modified Quick-Pulse Programming™ algorithm. It differs from older methods in the value used for  $V_{PP}$  (programming supply voltage) and in the width and number of the ALE/PROG pulses.

The 87C654 contains two signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes identify the device as an 87C654 manufactured by Philips Components.

Table 4 shows the logic levels for reading the signature byte, and for programming the program memory, the encryption table, and the lock bits. The circuit configuration and waveforms for quick-pulse programming are shown in Figures 12 and 13. Figure 14 shows the circuit configuration for normal program memory verification.

### Quick-Pulse Programming

The setup for microcontroller quick-pulse programming is shown in Figure 12. Note that the 87C654 is running with a 4 to 6MHz oscillator. The reason the oscillator needs to be running is that the device is executing internal address and program data transfers.

The address of the EPROM location to be programmed is applied to ports 1 and 2, as shown in Figure 12. The code byte to be programmed into that location is applied to port 0. RST, PSEN and pins of ports 2 and 3 specified in Table 4 are held at the 'Program Code Data' levels indicated in Table 4. The ALE/PROG is pulsed low 25 times as shown in Figure 13.

To program the encryption table, repeat the 25 pulse programming sequence for addresses 0 through 1FH, using the 'Pgm Encryption Table' levels. Do not forget that after the encryption table is programmed, verification cycles will produce only encrypted data.

To program the lock bits, repeat the 25 pulse programming sequence using the 'Pgm Lock Bit' levels. After one lock bit is programmed, further programming of the code memory and encryption table is disabled. However, the other lock bit can still be programmed.

Note that the EA/ $V_{PP}$  pin must not be allowed to go above the maximum specified  $V_{PP}$  level for any amount of time. Even a narrow glitch above that voltage can cause permanent damage to the device. The  $V_{PP}$  source should be well regulated and free of glitches and overshoot.

### Program Verification

If lock bit 2 has not been programmed, the on-chip program memory can be read out for program verification. The address of the program memory locations to be read is applied to ports 1 and 2 as shown in Figure 14. The other pins are held at the 'Verify Code Data' levels indicated in Table 4. The contents of the address location will be emitted on port 0. External pull-ups are required on port 0 for this operation.

If the encryption table has been programmed, the data presented at port 0 will be the exclusive NOR of the program byte with one of the encryption bytes. The user will have to know the encryption table contents in order to correctly decode the verification data. The encryption table itself cannot be read out.

### Reading the Signature Bytes

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. The values are:

(030H) = 15H indicates manufactured by Philips

(031H) = 99H indicates 87C654

### Program/Verify Algorithms

Any algorithm in agreement with the conditions listed in Table 4, and which satisfies the timing specifications, is suitable.

### Erase Characteristics

Erase of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent erasure. **For this and secondary effects, it is recommended that an opaque label be placed over the window.** For elevated temperature or environments where solvents are being used, apply Kapton tape Fluorglas part number 2345-5, or equivalent.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 angstroms) to an integrated dose of at least 15W-sec/cm<sup>2</sup>. Exposing the EPROM to an ultraviolet lamp of 12,000µW/cm<sup>2</sup> rating for 20 to 39 minutes, at a distance of about 1 inch, should be sufficient. Erasure leaves the array in an all 1s state.

Table 4. EPROM Programming Modes

MODE	RST	PSEN	ALE/PROG	EA/ $V_{PP}$	P2.7	P2.6	P3.7	P3.6
Read signature	1	0	1	1	0	0	0	0
Program code data	1	0	0*	$V_{PP}$	1	0	1	1
Verify code data	1	0	1	1	0	0	1	1
Pgm encryption table	1	0	0*	$V_{PP}$	1	0	1	0
Pgm lock bit 1	1	0	0*	$V_{PP}$	1	1	1	1
Pgm lock bit 2	1	0	0*	$V_{PP}$	1	1	0	0

#### NOTES:

1. '0' = Valid low for that pin, '1' = valid high for that pin.

2.  $V_{PP}$  = 12.75V  $\pm$  0.25V.

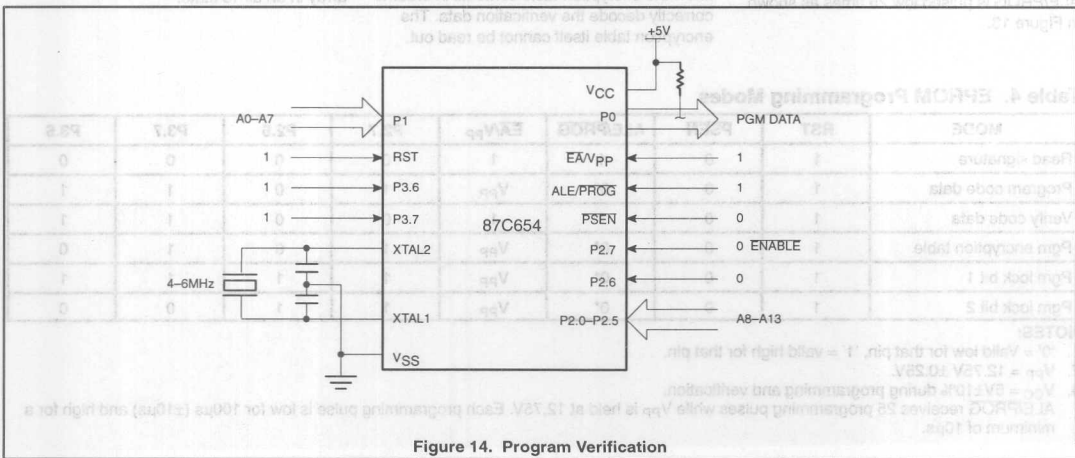
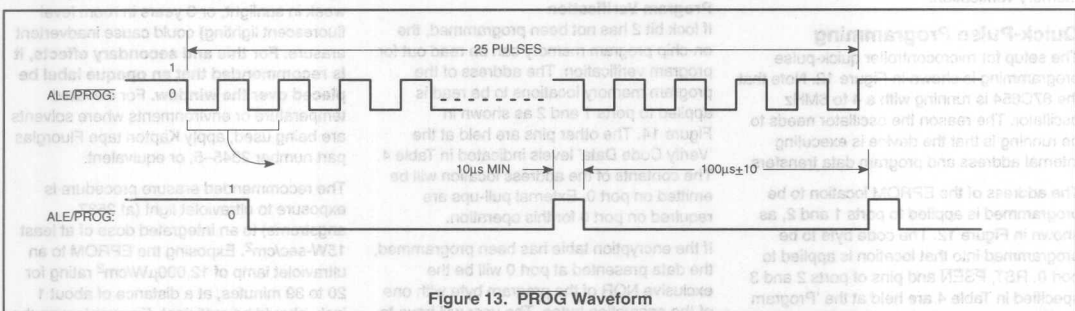
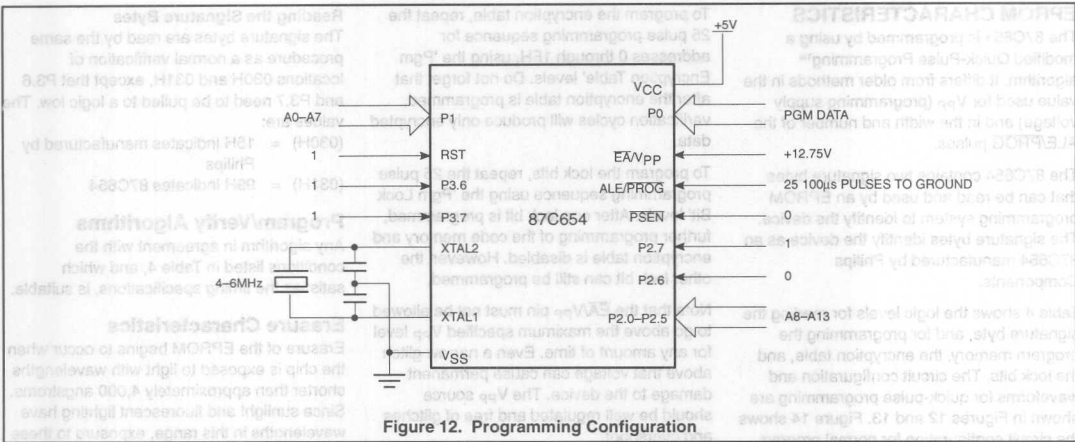
3.  $V_{CC}$  = 5V  $\pm$  10% during programming and verification.

\* ALE/PROG receives 25 programming pulses while  $V_{PP}$  is held at 12.75V. Each programming pulse is low for 100µs ( $\pm$ 10µs) and high for a minimum of 10µs.

™Trademark phrase of Intel Corporation.



# CMOS single-chip 8-bit microcontroller 87C654



## CMOS single-chip 8-bit microcontroller

87C654

## EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

 $T_{amb} = 21^{\circ}\text{C}$  to  $+27^{\circ}\text{C}$ ,  $V_{CC} = 5V \pm 10\%$ ,  $V_{SS} = 0V$  (See Figure 15)

SYMBOL	PARAMETER	MIN	MAX	UNIT
$V_{PP}$	Programming supply voltage	12.5	13.0	V
$I_{PP}$	Programming supply current		50	mA
$1/t_{CLCL}$	Oscillator frequency	4	6	MHz
$t_{AVGL}$	Address setup to $\overline{\text{PROG}}$ low	$48t_{CLCL}$		
$t_{GHAX}$	Address hold after $\overline{\text{PROG}}$	$48t_{CLCL}$		
$t_{DVGL}$	Data setup to $\overline{\text{PROG}}$ low	$48t_{CLCL}$		
$t_{GHDX}$	Data hold after $\overline{\text{PROG}}$	$48t_{CLCL}$		
$t_{EHS}$	P2.7 (ENABLE) high to $V_{PP}$	$48t_{CLCL}$		
$t_{SHGL}$	$V_{PP}$ setup to $\overline{\text{PROG}}$ low	10		$\mu\text{s}$
$t_{GHS}$	$V_{PP}$ hold after $\overline{\text{PROG}}$	10		$\mu\text{s}$
$t_{GLGH}$	$\overline{\text{PROG}}$ width	90	110	$\mu\text{s}$
$t_{AVQV}$	Address to data valid		$48t_{CLCL}$	
$t_{ELQZ}$	ENABLE low to data valid		$48t_{CLCL}$	
$t_{EHQZ}$	Data float after ENABLE	0	$48t_{CLCL}$	
$t_{GHGL}$	$\overline{\text{PROG}}$ high to $\overline{\text{PROG}}$ low	10		$\mu\text{s}$

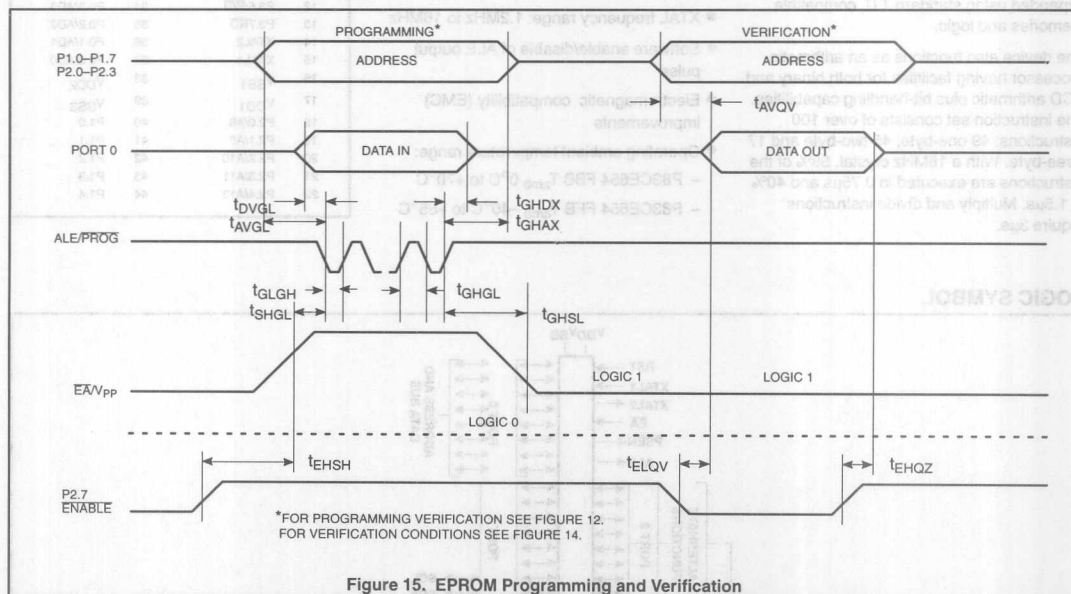


Figure 15. EPROM Programming and Verification



Purchase of Philips' I<sup>2</sup>C components conveys a license under the Philips' I<sup>2</sup>C patent to use the components in the I<sup>2</sup>C-system provided the system conforms to the I<sup>2</sup>C specifications defined by Philips.

# CMOS single-chip 8-bit microcontroller with Electromagnetic Compatibility improvements

**80CE654/83CE654**

## DESCRIPTION

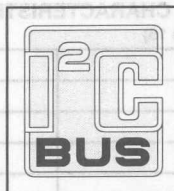
The 83CE654 Single-Chip 8-Bit Microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 83CE654 has the same instruction set as the 80C51. Two versions of the derivative exist:

83CE654 — 16k bytes mask programmable ROM, 256 bytes RAM

80CE654 — ROMless version of the 83CE654

This device provides architectural enhancements that make it applicable in a variety of applications for general control systems. The 8XCE654 contains a non-volatile 16k × 8 read-only program memory (83CE654), a volatile 256 × 8 read/write data memory, four 8-bit I/O ports, two 16-bit timer/event counters (identical to the timers of the 80C51), a multi-source, two-priority-level, nested interrupt structure, an I<sup>2</sup>C interface, UART and on-chip oscillator and timing circuits. For systems that require extra capability, the 8XCE654 can be expanded using standard TTL compatible memories and logic.

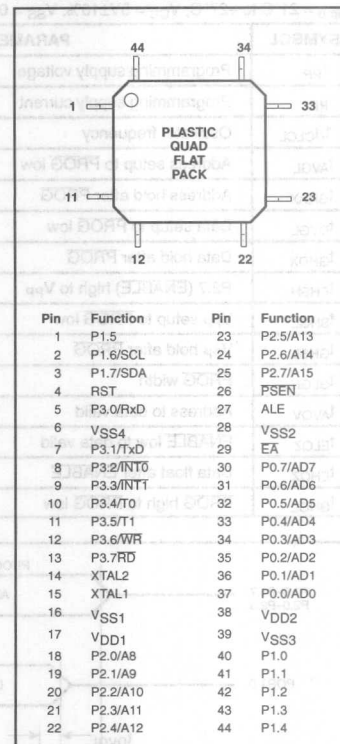
The device also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of over 100 instructions: 49 one-byte, 45 two-byte and 17 three-byte. With a 16MHz crystal, 58% of the instructions are executed in 0.75µs and 40% in 1.5µs. Multiply and divide instructions require 3µs.



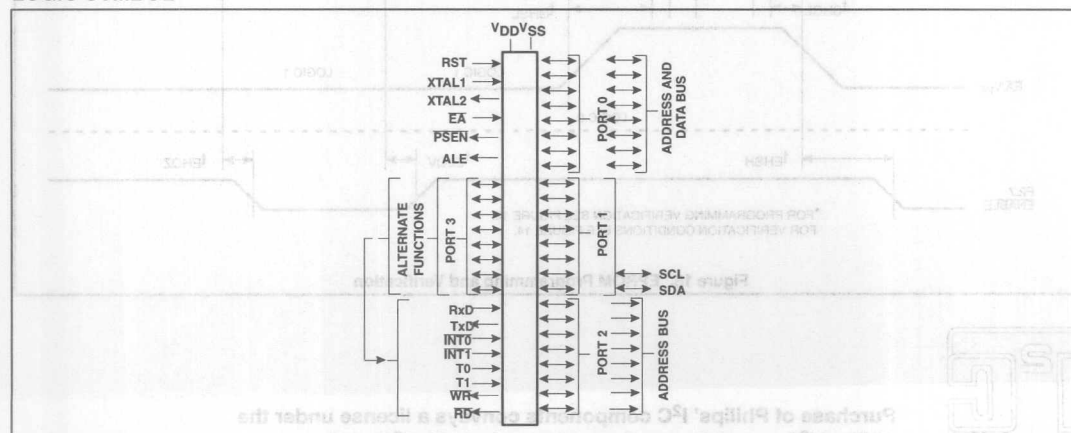
## FEATURES

- 80C51 central processing unit
- 16k × 8 ROM expandable externally to 64k bytes
- 256 × 8 RAM, expandable externally to 64k bytes
- Two standard 16-bit timer/counters
- Four 8-bit I/O ports
- I<sup>2</sup>C-bus serial I/O port with byte oriented master and slave functions
- Full-duplex UART facilities
- ROM code protection
- XTAL frequency range: 1.2MHz to 16MHz
- Software enable/disable of ALE output pulse
- Electromagnetic compatibility (EMC) improvements
- Operating ambient temperature range:
  - P83CE654 FBB T<sub>amb</sub> 0°C to +70°C
  - P83CE654 FFB T<sub>amb</sub> -40°C to +85°C

## PIN CONFIGURATION



## LOGIC SYMBOL



# CMOS single-chip 8-bit microcontroller with Electromagnetic Compatibility improvements

## 80CE654/83CE654

### ORDERING INFORMATION

ROMless	ROM	TEMPERATURE RANGE °C AND PACKAGE	FREQUENCY MHz	DRAWING NUMBER
P80CE654FBB	P83CE654FBB	0 to +70, Plastic Quad Flat Pack	1.2 to 16	SOT307-2 <sup>1</sup>
P80CE654FFB	P83CE654FFB	-40 to +85, Plastic Quad Flat Pack	1.2 to 16	SOT307-2 <sup>1</sup>

#### NOTE:

1. SOT311 replaced by SOT307-2.

### ELECTROMAGNETIC COMPATIBILITY (EMC) IMPROVEMENTS

Primary attention is paid on the reduction of electromagnetic emission of the microcontroller P83CE654.

The following features effect in reducing the electromagnetic emission and additionally improve the electromagnetic susceptibility:

- Two supply voltage pins ( $V_{DD1}$ ,  $V_{DD2}$ ) and four ground pins ( $V_{SS1}$  to  $V_{SS4}$ )
- Separate  $V_{DD}$  pins for the internal logic and the port buffers

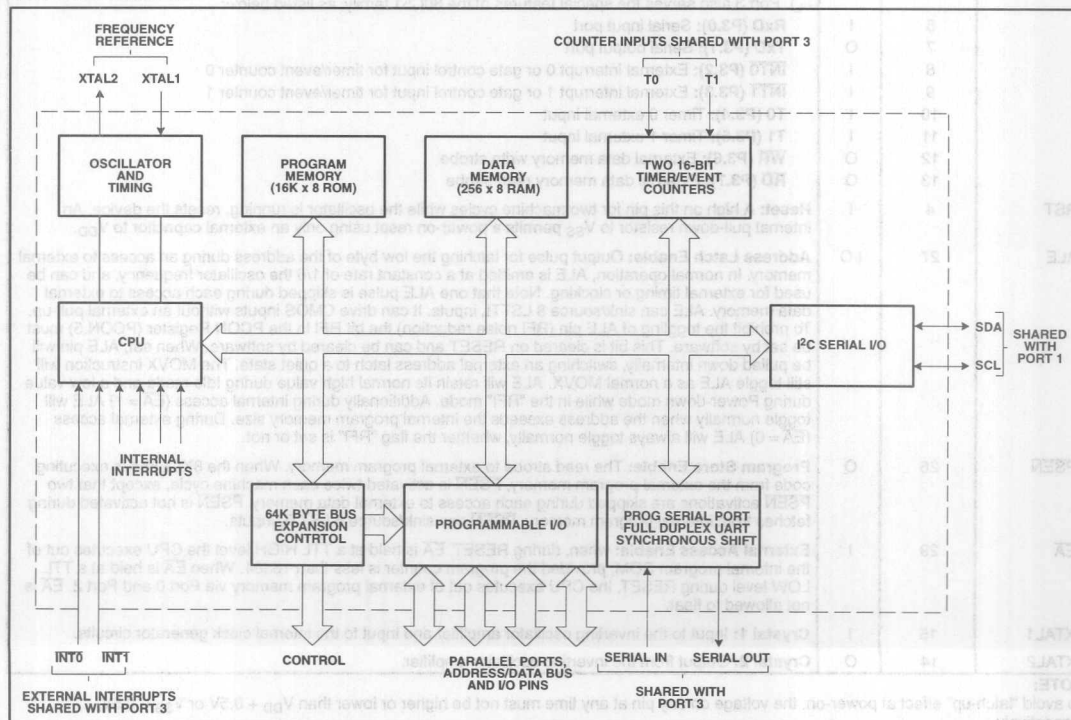
- Internal decoupling capacitance improves the EMC radiation behavior and the EMC immunity

- External capacitors are to be located as close as possible between pins  $V_{DD2}$  and  $V_{SS3}$  as well as  $V_{DD1}$  and  $V_{SS1}$ ; ceramic chip capacitors are recommended (100nF).

- The ALE output signal (pulses at a frequency of  $f_{OSC}/6$ ) can be disabled under software control (bit 5 in the SFR PCON: "RFI"); if disabled, no ALE pulse will occur.

ALE pin will be pulled down internally, switching an external address latch to a quiet state. The MOVX instruction will still toggle ALE as a normal MOVX. ALE will retain its normal high value during Idle mode and a low value during Power-down mode while in the "RFI" reduction mode. Additionally during internal access ( $EA = 1$ ) ALE will toggle normally when the address exceeds the internal program memory size. During external access ( $EA = 0$ ) ALE will always toggle normally, whether the flag "RFI" is set or not.

### BLOCK DIAGRAM



# CMOS single-chip 8-bit microcontroller with Electromagnetic Compatibility improvements

80CE654/83CE654

## PIN DESCRIPTIONS

MNEMONIC	PIN NUMBER	TYPE	NAME AND FUNCTION
V <sub>SS1</sub> , V <sub>SS2</sub> , V <sub>SS3</sub> , V <sub>SS4</sub>	16, 28, 39, 6	I	<b>Ground:</b> 0V reference. All pins must be connected.
V <sub>DD1</sub> , V <sub>DD2</sub>	17, 38	I	<b>Power Supply:</b> This is the power supply voltage for normal, idle, and power-down operation. Both pins must be connected.
P0.0–0.7	37–30	I/O	<b>Port 0:</b> Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s. Port 0 can sink/source 8 LSTTL inputs.
P1.0–P1.7	40–44, 1–3	I/O	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pull-ups, except P1.6 and P1.7 which are open drain. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 1 also receives the low-order address byte during program memory verification. Alternate functions include: <b>SCL:</b> I <sup>2</sup> C-bus serial port clock line. <b>SDA:</b> I <sup>2</sup> C-bus serial port data line.
P1.6	2	I/O	
P1.7	3	I/O	
P2.0–P2.7	18–25	I/O	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register.
P3.0–P3.7	5, 7–13	I/O	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 3 also serves the special features of the 80C51 family, as listed below: <b>RxD (P3.0):</b> Serial input port <b>TxD (P3.1):</b> Serial output port <b>INT0 (P3.2):</b> External interrupt 0 or gate control input for timer/event counter 0 <b>INT1 (P3.3):</b> External interrupt 1 or gate control input for timer/event counter 1 <b>T0 (P3.4):</b> Timer 0 external input <b>T1 (P3.5):</b> Timer 1 external input <b>WR (P3.6):</b> External data memory write strobe <b>RD (P3.7):</b> External data memory read strobe
RST	4	I	<b>Reset:</b> A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal pull-down resistor to V <sub>SS</sub> permits a power-on reset using only an external capacitor to V <sub>DD</sub> .
ALE	27	I/O	<b>Address Latch Enable:</b> Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory. ALE can sink/source 8 LSTTL inputs. It can drive CMOS inputs without an external pull-up. To prohibit the toggling of ALE pin (RFI noise reduction) the bit RFI in the PCON Register (PCON.5) must be set by software. This bit is cleared on RESET and can be cleared by software. When set, ALE pin will be pulled down internally, switching an external address latch to a quiet state. The MOVX instruction will still toggle ALE as a normal MOVX. ALE will retain its normal high value during Idle mode and a low value during Power-down mode while in the "RFI" mode. Additionally during internal access (EA = 1) ALE will toggle normally when the address exceeds the internal program memory size. During external access (EA = 0) ALE will always toggle normally, whether the flag "RFI" is set or not.
PSEN	26	O	<b>Program Store Enable:</b> The read strobe to external program memory. When the 8XCE654 is executing code from the external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory. PSEN is not activated during fetches from internal program memory. PSEN can sink/source 8 LSTTL inputs.
EA	29	I	<b>External Access Enable:</b> when, during RESET, EA is held at a TTL HIGH level the CPU executes out of the internal program ROM, provided the program counter is less than 16384. When EA is held at a TTL LOW level during RESET, the CPU executes out of external program memory via Port 0 and Port 2. EA is not allowed to float.
XTAL1	15	I	<b>Crystal 1:</b> Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	14	O	<b>Crystal 2:</b> Output from the inverting oscillator amplifier.

### NOTE:

To avoid "latch-up" effect at power-on, the voltage on any pin at any time must not be higher or lower than V<sub>DD</sub> + 0.5V or V<sub>SS</sub> - 0.5V, respectively.



# CMOS single-chip 8-bit microcontroller with Electromagnetic Compatibility improvements

## 80CE654/83CE654

Table 1. 8XCE654 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB							LSB	
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
DPTR:	Data pointer (2 bytes)										
DPH	Data pointer high	83H									00H
DPL	Data pointer low	82H									00H
IE*#	Interrupt enable	A8H	AF	AE	AD	AC	AB	AA	A9	A8	0x000000B
			EA		ES1	ES0	ET1	EX1	ET0	EX0	
			BF	BE	BD	BC	BB	BA	B9	B8	
IP*#	Interrupt priority	B8H	—		PS1	PS0	PT1	PX1	PT0	PX0	xx000000B
			87	86	85	84	83	82	81	80	
P0*	Port 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
			97	96	95	94	93	92	91	90	
P1*#	Port 1	90H	SDA	SCL							FFH
			A7	A6	A5	A4	A3	A2	A1	A0	
P2*	Port 2	A0H	A15	A14	A13	A12	A11	A10	A9	A8	FFH
			B7	B6	B5	B4	B3	B2	B1	B0	
P3*	Port 3	B0H	RD	WR	T1	T0	INT1	INT0	TXD	RXD	FFH
PCON	Power control	87H	SMOD	—	RF1	—	GF1	GF0	PD	IDL	0xxx0000B
			9F	9E	9D	9C	9B	9A	99	98	
S0CON*#	Serial 0 port control	98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00H
S0BUF#	Serial 0 data buffer	99H									xxxxxxxB
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program status word	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	00H
S1DAT#	Serial 1 data	DAH									00H
SP	Stack pointer	81H									07H
S1ADR#	Serial 1 address	DBH	SLAVE ADDRESS								00H
S1STA#	Serial 1 status	D9H	SC4	SC3	SC2	SC1	SC0	0	0	0	FFH
			DF	DE	DD	DC	DB	DA	D9	D8	
S1CON*#	Serial 1 control	D8H	CR2	ENS1	STA	STO	SI	AA	CR1	CR0	00000000B
			8F	8E	8D	8C	8B	8A	89	88	
TCON*	Timer control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
TH1	Timer high 1	8DH									00H
TH0	Timer high 0	8CH									00H
TL1	Timer low 1	8BH									00H
TH0	Timer low 0	8AH									00H
TMOD	Timer mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.

## CMOS single-chip 8-bit microcontroller with Electromagnetic Compatibility improvements

80CE654/83CE654

### ROM CODE PROTECTION

#### (83CE654)

The 83CE654 has an additional security feature. ROM code protection may be selected by setting a mask-programmable security bit (i.e., user dependent). This feature may be requested during ROM code submission. When selected, the ROM code is protected and cannot be read out at any time by any test mode or by any instruction in the external program memory space.

The MOV<sub>C</sub> instructions are the only instructions that have access to program code in the internal or external program memory. The EA input is latched during RESET and is "don't care" after RESET (also if the security bit is not set). This implementation prevents reading internal program code by switching from external program memory to internal program memory during a MOV<sub>C</sub> instruction or any other instruction that uses immediate data.

Table 2 lists the access to the internal and external program memory by the MOV<sub>C</sub> instructions when the security bit has been set to a logical "1".

### OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator, as shown in the Logic Symbol, page 992.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

### Reset

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. To insure a good power-on reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few milliseconds) plus two machine cycles. At power-on, the voltage on V<sub>DD</sub> and RST must come up at the same time for a proper start-up.

### Power-on Reset (See Figure 1.)

When V<sub>DD</sub> is turned on, and provided its rise-time does not exceed 10ms, an automatic reset can be obtained by connecting the RST pin to V<sub>DD</sub> via a 2.2μF capacitor. When the power is switched on, the voltage on the RST pin is equal to V<sub>DD</sub> minus the capacitor voltage, and decreases from V<sub>DD</sub> as the capacitor charges through the internal resistor (R<sub>RST</sub>) to ground. The larger the capacitor, the more slowly V<sub>RST</sub> decreases. V<sub>RST</sub> must remain above the lower threshold of the Schmitt trigger long enough to effect a complete reset. The time required is the oscillator start-up time, plus 2 machine cycles.

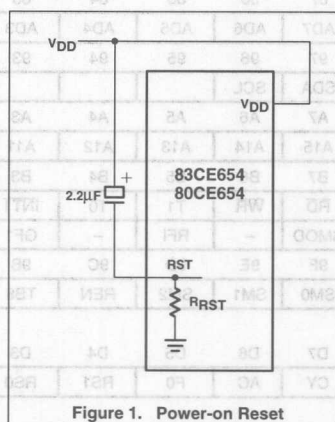


Figure 1. Power-on Reset

### Idle Mode

In the idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

### Power-Down Mode

In the power-down mode, the oscillator is stopped and the instruction to invoke power-down is the last instruction executed. Only the contents of the on-chip RAM are preserved. A hardware reset is the only way

to terminate the power-down mode. The control bits for the reduced power modes are in the special function register PCON. Table 3 shows the state of the I/O ports during low current operating modes.

### Power Control Register PCON

These special modes are activated by software via the Special Function Register PCON. Its hardware address is 87H. PCON is not bit addressable. The reset value of PCON is (0x0x0000).

Bit	Symbol	Function
PCON.7	SMOD	Double Baud rate bit. When set to logic 1 the baud rate is doubled when Timer 1 is used to generate baud rate, and the Serial Port is used in modes 1, 2 or 3.
PCON.6	—	(reserved for future use*)
PCON.5	RFI	When set to logic 1 the toggling of ALE pin is prohibited. This bit is cleared on RESET.
PCON.4	—	(reserved for future use*)
PCON.3	GF1	General purpose flag bit.
PCON.2	GF0	General purpose flag bit.
PCON.1	PD	Power-down bit. Setting this bit activates Power-down mode.
PCON.0	IDL	Idle mode bit. Setting this bit activates the Idle mode. If 1s are written to PD and IDL at the same time, PD takes precedence.

#### NOTE:

- \* User software should not write 1s to reserved bits. These bits may be used in future 80C51 family products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1. The value read from a reserved bit is indeterminate.

### I<sup>2</sup>C Serial Communication—SIO1

The I<sup>2</sup>C serial port is identical to the I<sup>2</sup>C serial port on the 8XC52. The operation of this subsystem is described in detail in the 8XC52 section of this manual.

Note that in both the 8XC652/4 and the 8XC52 the I<sup>2</sup>C pins are alternate functions to port pins P1.6 and P1.7. Because of this, P1.6 and P1.7 on these parts do not have a pull-up structure as found on the 80C51. Therefore P1.6 and P1.7 have open drain outputs on the 8XC652/4.

CMOS single-chip 8-bit microcontroller with  
Electromagnetic Compatibility improvements

80CE654/83CE654

Table 2.

	ACCESS TO INTERNAL PROGRAM MEMORY	ACCESS TO EXTERNAL PROGRAM MEMORY
MOV <sub>C</sub> in internal program memory	YES	YES
MOV <sub>C</sub> in external program memory	NO	YES

NOTE:

If the security bit has been set to a logical 0, there are no restrictions for the MOV<sub>C</sub> instructions.

Table 3. External Pin Status During Idle and Power-Down Mode

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

Serial Control Register (S1CON) – See Table 4

S1CON (D8H)	CR2	ENS1	STA	STO	SI	AA	CR1	CR0
-------------	-----	------	-----	-----	----	----	-----	-----

Bits CR0, CR1 and CR2 determine the serial clock frequency that is generated in the master mode of operation.

Table 4. Serial Clock Rates

CR2	CR1	CR0	BIT FREQUENCY (kHz) AT f <sub>osc</sub>			
			6MHz	12MHz	16MHz	f <sub>osc</sub> DIVIDED BY
0	0	0	23	47	63	256
0	0	1	27	54	71	224
0	1	0	31	63	83	192
0	1	1	37	75	100	160
1	0	0	6.25	12.5	17	960
1	0	1	50	100	133 <sup>1</sup>	120
1	1	0	100	200 <sup>1</sup>	267 <sup>1</sup>	60
1	1	1	0.24 < 62.5	0.49 < 62.5	0.65 < 55.6	96 × (256 – (reload value Timer 1)) reload value range Timer 1 (in mode 2)
			0 < 255	0 < 254	0 < 253	

NOTES:

1. These frequencies exceed the upper limit of 100kHz of the I<sup>2</sup>C-bus specification and cannot be used in an I<sup>2</sup>C-bus application.

CMOS single-chip 8-bit microcontroller with  
Electromagnetic Compatibility improvements

80CE654/83CE654

ABSOLUTE MAXIMUM RATINGS

PARAMETER	RATING	UNIT
Voltage on V <sub>DD</sub> to V <sub>SS</sub>	−0.5 to +6.5	V
Voltage on any pin to V <sub>SS</sub>	−0.5 to V <sub>DD</sub> +0.5	V
Storage temperature range	−65 to +150	°C
Power dissipation (based on package heat transfer limitations, not device power consumption) <sup>1</sup>	1	W
Operating ambient temperature range		
FBB	0 to +70	°C
FFB	−40 to +85	°C

NOTE:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device, this is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied. Exposure to Absolute Maximum Rating conditions for extended periods may affect device reliability.

DEVICE SPECIFICATIONS

TYPE	SUPPLY VOLTAGE (V)		FREQUENCY (MHz)		TEMPERATURE RANGE (°C)
	MIN.	MAX.	MIN.	MAX.	
P83(0)CE654FBB	4.5	5.5	1.2	16	0 to +70
P83(0)CE654FFB	4.5	5.5	1.2	16	−40 to +85

CLOCK	CLOCK	CLOCK	BIT FREQUENCY (kHz) AT 10°C		
			16MHz	12MHz	8MHz
0	0	0	0	0	0
1	1	1	1	1	1
2	2	2	2	2	2
3	3	3	3	3	3
4	4	4	4	4	4
5	5	5	5	5	5
6	6	6	6	6	6
7	7	7	7	7	7
8	8	8	8	8	8
9	9	9	9	9	9
10	10	10	10	10	10
11	11	11	11	11	11
12	12	12	12	12	12
13	13	13	13	13	13
14	14	14	14	14	14
15	15	15	15	15	15
16	16	16	16	16	16
17	17	17	17	17	17
18	18	18	18	18	18
19	19	19	19	19	19
20	20	20	20	20	20
21	21	21	21	21	21
22	22	22	22	22	22
23	23	23	23	23	23
24	24	24	24	24	24
25	25	25	25	25	25
26	26	26	26	26	26
27	27	27	27	27	27
28	28	28	28	28	28
29	29	29	29	29	29
30	30	30	30	30	30
31	31	31	31	31	31
32	32	32	32	32	32
33	33	33	33	33	33
34	34	34	34	34	34
35	35	35	35	35	35
36	36	36	36	36	36
37	37	37	37	37	37
38	38	38	38	38	38
39	39	39	39	39	39
40	40	40	40	40	40
41	41	41	41	41	41
42	42	42	42	42	42
43	43	43	43	43	43
44	44	44	44	44	44
45	45	45	45	45	45
46	46	46	46	46	46
47	47	47	47	47	47
48	48	48	48	48	48
49	49	49	49	49	49
50	50	50	50	50	50
51	51	51	51	51	51
52	52	52	52	52	52
53	53	53	53	53	53
54	54	54	54	54	54
55	55	55	55	55	55
56	56	56	56	56	56
57	57	57	57	57	57
58	58	58	58	58	58
59	59	59	59	59	59
60	60	60	60	60	60
61	61	61	61	61	61
62	62	62	62	62	62
63	63	63	63	63	63
64	64	64	64	64	64
65	65	65	65	65	65
66	66	66	66	66	66
67	67	67	67	67	67
68	68	68	68	68	68
69	69	69	69	69	69
70	70	70	70	70	70
71	71	71	71	71	71
72	72	72	72	72	72
73	73	73	73	73	73
74	74	74	74	74	74
75	75	75	75	75	75
76	76	76	76	76	76
77	77	77	77	77	77
78	78	78	78	78	78
79	79	79	79	79	79
80	80	80	80	80	80
81	81	81	81	81	81
82	82	82	82	82	82
83	83	83	83	83	83
84	84	84	84	84	84
85	85	85	85	85	85
86	86	86	86	86	86
87	87	87	87	87	87
88	88	88	88	88	88
89	89	89	89	89	89
90	90	90	90	90	90
91	91	91	91	91	91
92	92	92	92	92	92
93	93	93	93	93	93
94	94	94	94	94	94
95	95	95	95	95	95
96	96	96	96	96	96
97	97	97	97	97	97
98	98	98	98	98	98
99	99	99	99	99	99
100	100	100	100	100	100
101	101	101	101	101	101
102	102	102	102	102	102
103	103	103	103	103	103
104	104	104	104	104	104
105	105	105	105	105	105
106	106	106	106	106	106
107	107	107	107	107	107
108	108	108	108	108	108
109	109	109	109	109	109
110	110	110	110	110	110
111	111	111	111	111	111
112	112	112	112	112	112
113	113	113	113	113	113
114	114	114	114	114	114
115	115	115	115	115	115
116	116	116	116	116	116
117	117	117	117	117	117
118	118	118	118	118	118
119	119	119	119	119	119
120	120	120	120	120	120
121	121	121	121	121	121
122	122	122	122	122	122
123	123	123	123	123	123
124	124	124	124	124	124
125	125	125	125	125	125
126	126	126	126	126	126
127	127	127	127	127	127
128	128	128	128	128	128
129	129	129	129	129	129
130	130	130	130	130	130
131	131	131	131	131	131
132	132	132	132	132	132
133	133	133	133	133	133
134	134	134	134	134	134
135	135	135	135	135	135
136	136	136	136	136	136
137	137	137	137	137	137
138	138	138	138	138	138
139	139	139	139	139	139
140	140	140	140	140	140
141	141	141	141	141	141
142	142	142	142	142	142
143	143	143	143	143	143
144	144	144	144	144	144
145	145	145	145	145	145
146	146	146	146	146	146
147	147	147	147	147	147
148	148	148	148	148	148
149	149	149	149	149	149
150	150	150	150	150	150
151	151	151	151	151	151
152	152	152	152	152	152
153	153	153	153	153	153
154	154	154	154	154	154
155	155	155	155	155	155
156	156	156	156	156	156
157	157	157	157	157	157
158	158	158	158	158	158
159	159	159	159	159	159
160	160	160	160	160	160
161	161	161	161	161	161
162	162	162	162	162	162
163	163	163	163	163	163
164	164	164	164	164	164
165	165	165	165	165	165
166	166	166	166	166	166
167	167	167	167	167	167
168	168	168	168	168	168
169	169	169	169	169	169
170	170	170	170	170	170
171	171	171	171	171	171
172	172	172	172	172	172
173	173	173	173	173	173
174	174	174	174	174	174
175	175	175	175	175	175
176	176	176	176	176	176
177	177	177	177	177	177
178	178	178	178	178	178
179	179	179	179	179	179
180	180	180	180	180	180
181	181	181	181	181	181
182	182	182	182	182	182
183	183	183	183	183	183
184	184	184	184	184	184
185	185	185	185	185	185
186	186	186	186	186	186
187	187	187	187	187	187
188	188	188	188	188	188
189	189	189	189	189	189
190	190	190	190	190	190
191	191	191	191	191	191
192	192	192	192	192	192
193	193	193	193	193	193
194	194	194	194	194	194
195	195	195	195	195	195
196	196	196	196	196	196
197	197	197	197	197	197
198	198	198	198	198	198
199	199	199	199	199	199
200	200	200	200	200	200
201	201	201	201	201	201
202	202	202	202	202	202
203	203	203	203	203	203
204	204	204	204	204	204
205	205	205	205	205	205
206	206	206	206	206	206
207	207	207	207	207	207
208	208	208	208	208	208
209	209	209	209	209	209
210	210	210	210	210	210
211	211	211	211	211	211
212	212	212	212	212	212
213	213	213	213	213	213
214	214	214	214	214	214
215	215	215	215	215	215
216	216	216	216	216	216
217	217	217	217	217	217
218	218	218	218	218	218
219	219	219	219	219	219
220	220	220	220	220	220
221	221	221	221	221	221
222	222	222	222	222	222
223	223	223	223	223	223
224	224	224	224	224	224
225	225	225	225	225	225
226	226	226	226	226	226
227	227	227	227	227	227
228	228	228	228	228	228

# CMOS single-chip 8-bit microcontroller with Electromagnetic Compatibility improvements

80CE654/83CE654

## DC ELECTRICAL CHARACTERISTICS

 $V_{DD} = 5V (\pm 10\%), V_{SS} = 0V, T_{amb} = 0^\circ C \text{ to } +70^\circ C \text{ or } -40^\circ C \text{ to } +85^\circ C$ 

SYMBOL	PARAMETER	PART TYPE	TEST CONDITIONS	LIMITS		UNIT
				MIN.	MAX.	
$V_{IL}$	Input low voltage, except EA, P1.6/SCL, P1.7/SDA	0 to +70°C -40 to +85°C		-0.5 -0.5	$0.2V_{DD}-0.1$ $0.2V_{DD}-0.15$	V V
$V_{IL1}$	Input low voltage to EA	0 to +70°C -40 to +85°C		-0.5 -0.5	$0.2V_{DD}-0.3$ $0.2V_{DD}-0.35$	V V
$V_{IL2}$	Input low voltage to P1.6/SCL, P1.7/SDA <sup>6</sup>			-0.5	$0.3V_{DD}$	V
$V_{IH}$	Input high voltage, except XTAL1, RST, P1.6/SCL, P1.7/SDA	0 to +70°C -40 to +85°C		$0.2V_{DD}+0.9$ $0.2V_{DD}+1.0$	$V_{DD}+0.5$ $V_{DD}+0.5$	V V
$V_{IH1}$	Input high voltage, XTAL1, RST	0 to +70°C -40 to +85°C		$0.7V_{DD}$ $0.7V_{DD}+0.1$	$V_{DD}+0.5$ $V_{DD}+0.5$	V V
$V_{IH2}$	Input high voltage, P1.6/SCL, P1.7/SDA <sup>6</sup>			$0.7V_{DD}$	6.0	V
$V_{OL}$	Output low voltage, ports 1, 2, 3, except P1.6/SCL, P1.7/SDA <sup>4</sup> , ALE, PSEN		$I_{OL} = 1.6mA^7$		0.45	V
$V_{OL1}$	Output low voltage, port 0, ALE, PSEN <sup>4</sup>		$I_{OL} = 3.2mA^7$		0.45	V
$V_{OL2}$	Output low voltage, P1.6/SCL, P1.7/SDA <sup>4</sup>		$I_{OL} = 3.0mA^7$		0.4	V
$V_{OH}$	Output high voltage, ports 1, 2, 3, except P1.6, P1.7, ALE, PSEN		$I_{OH} = -60\mu A$ ; $V_{DD} = 5V (\pm 10\%)$ $I_{OH} = -25\mu A$ $I_{OH} = -10\mu A$	2.4  $0.75V_{DD}$ $0.9V_{DD}$		V V V
$V_{OH1}$	Output high voltage; port 0 in external bus mode <sup>5</sup>		$I_{OH} = -800\mu A$ ; $V_{DD} = 5V (\pm 10\%)$ $I_{OH} = -300\mu A$ $I_{OH} = -80\mu A$	2.4  $0.75V_{DD}$ $0.9V_{DD}$		V V V
$I_{IL}$	Logical 0 input current, ports 1, 2, 3, except P1.6/SCL, P1.7/SDA	0 to +70°C -40 to +85°C	$V_i = 0.45V$ $V_i = 0.45V$		-50 -75	$\mu A$ $\mu A$
$I_{TL}$	Logical 1-to-0 transition current, ports 1, 2, 3, except P1.6/SCL, P1.7/SDA	0 to +70°C -40 to +85°C	$V_i = 2.0V$ $V_i = 2.0V$		-650 -750	$\mu A$ $\mu A$
$I_{L11}$	Input leakage current, port 0, EA		$0.45V < V_i < V_{DD}$		$\pm 10$	$\mu A$
$I_{L12}$	Input leakage current, P1.6/SCL, P1.7/SDA		$0V < V_i < 5.5V$ $0V < V_{DD} < 5.5V$		$\pm 10$	$\mu A$
$I_{DD}$	Power supply current: Active mode @ 16MHz <sup>1, 8</sup> Idle mode @ 16MHz <sup>2, 8</sup> Power down mode <sup>3</sup>		$V_{DD} = 5.5V$ $V_{DD} = 5V \pm 10\%$ $@ 2V < V_{PD} < V_{DDMAX}$		22 6 50	mA mA $\mu A$
$R_{RST}$	Internal reset pull-down resistor			50	150	k $\Omega$
$C_{IO}$	Pin capacitance of I/O buffer		Freq.=1MHz; $T_{amb} = 25^\circ C$		10	pF

NOTES: See Next Page.



CMOS single-chip 8-bit microcontroller with  
Electromagnetic Compatibility improvements

80CE654/83CE654

NOTES FOR DC ELECTRICAL CHARACTERISTICS:

1. The operating supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 5\text{ns}$ ;  $V_{IL} = V_{SS} + 0.5\text{V}$ ;  $V_{IH} = V_{DD} - 0.5\text{V}$ ; XTAL2 not connected; EA = RST = Port 0 = P1.6 = P1.7 =  $V_{DD}$ .
2. The idle mode supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 5\text{ns}$ ;  $V_{IL} = V_{SS} + 0.5\text{V}$ ;  $V_{IH} = V_{DD} - 0.5\text{V}$ ; XTAL2 not connected; Port 0 = P1.6 = P1.7 =  $V_{DD}$ ; EA = RST =  $V_{SS}$ .
3. The power-down current is measured with all output pins disconnected; XTAL2 not connected; Port 0 = P1.6 = P1.7 =  $V_{DD}$ ; EA = XTAL1 = RST =  $V_{SS}$ .
4. Capacitive loading on ports 0 and 2 may cause spurious noise to be superimposed on the  $V_{OL}$ s of ALE and ports 1 and 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading  $> 100\text{pF}$ ), the noise pulse on the ALE pin may exceed  $0.8\text{V}$ . In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.
5. Capacitive loading on ports 0 and 2 may cause the  $V_{OH}$  on ALE and PSEN to momentarily fall below the  $0.9V_{DD}$  specification when the address bits are stabilizing.
6. The input threshold voltage of P1.6 and P1.7 (SIO1) meets the I<sup>2</sup>C specification, so an input voltage below  $0.3V_{DD}$  will be recognized as a logic 0 while an input voltage above  $0.7V_{DD}$  will be recognized as a logic 1.
7. Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows: Maximum  $I_{OL} = 10\text{mA}$  per port pin; Maximum  $I_{OL} = 26\text{mA}$  total for Port 0; Maximum  $I_{OL} = 15\text{mA}$  total for Ports 1, 2, and 3; Maximum  $I_{OL} = 71\text{mA}$  total for all output pins. If  $I_{OL}$  exceeds the test conditions,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
8.  $I_{DD\text{MAX}}$  for the 80/83CE654 at the other frequencies can be derived from Figure 1, where FREQ is the external oscillator frequency in MHz.  $I_{DD\text{MAX}}$  is given in mA.

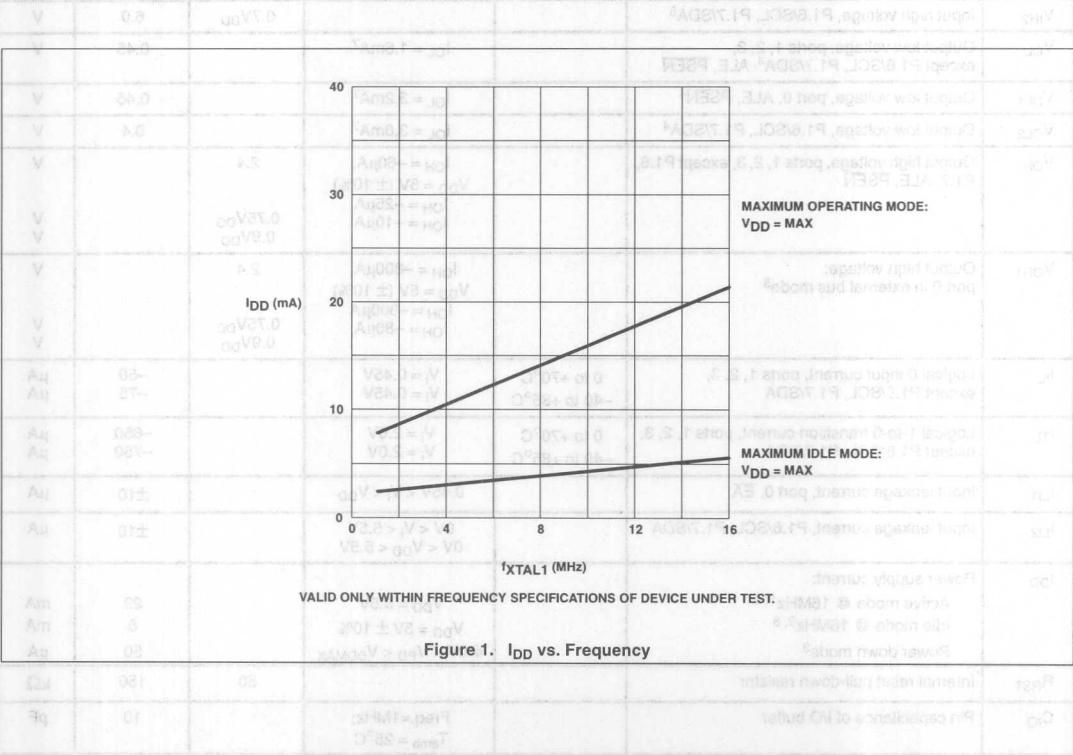


Figure 1.  $I_{DD}$  vs. Frequency

# CMOS single-chip 8-bit microcontroller with Electromagnetic Compatibility improvements

80CE654/83CE654

## AC ELECTRICAL CHARACTERISTICS<sup>1, 2</sup>

SYMBOL	FIGURE	PARAMETER	16MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
$t_{CLCL}$	2	Oscillator frequency			1.2	16	MHz
$t_{LHLL}$	2	ALE pulse width	85		$2t_{CLCL}-40$		ns
$t_{AVLL}$	2	Address valid to ALE low	8		$t_{CLCL}-55$		ns
$t_{LLAX}$	2	Address hold after ALE low	28		$t_{CLCL}-35$		ns
$t_{LLIV}$	2	ALE low to valid instruction in		150		$4t_{CLCL}-100$	ns
$t_{LLPL}$	2	ALE low to PSEN low	23		$t_{CLCL}-40$		ns
$t_{PLPH}$	2	PSEN pulse width	143		$3t_{CLCL}-45$		ns
$t_{PLIV}$	2	PSEN low to valid instruction in		83		$3t_{CLCL}-105$	ns
$t_{PXIX}$	2	Input instruction hold after PSEN	0		0		ns
$t_{PXIZ}$	2	Input instruction float after PSEN		38		$t_{CLCL}-25$	ns
$t_{AVIV}$	2	Address to valid instruction in		208		$5t_{CLCL}-105$	ns
$t_{PLAZ}$	2	PSEN low to address float		10		10	ns
<b>Data Memory</b>							
$t_{AVLL}$	3, 4	Address valid to ALE low	8		$t_{CLCL}-55$		ns
$t_{RLRH}$	3, 4	RD pulse width	275		$6t_{CLCL}-100$		ns
$t_{WLWH}$	3, 4	WR pulse width	275		$6t_{CLCL}-100$		ns
$t_{RLDV}$	3, 4	RD low to valid data in		148		$5t_{CLCL}-165$	ns
$t_{RHDX}$	3, 4	Data hold after RD	0		0		ns
$t_{RHDZ}$	3, 4	Data float after RD		55		$2t_{CLCL}-70$	ns
$t_{LLDV}$	3, 4	ALE low to valid data in		350		$8t_{CLCL}-150$	ns
$t_{AVDV}$	3, 4	Address to valid data in		398		$9t_{CLCL}-165$	ns
$t_{LLWL}$	3, 4	ALE low to RD or WR low	138	238	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
$t_{AVWL}$	3, 4	Address valid to WR low or RD low	120		$4t_{CLCL}-130$		ns
$t_{QVWX}$	3, 4	Data valid to WR transition	3		$t_{CLCL}-60$		ns
$t_{DW}$	3, 4	Data setup time before WR	288		$7t_{CLCL}-150$		ns
$t_{WHQX}$	3, 4	Data hold after WR	13		$t_{CLCL}-50$		ns
$t_{RLAZ}$	3, 4	RD low to address float		0		0	ns
$t_{WLHL}$	3, 4	RD or WR high to ALE high	23	103	$t_{CLCL}-40$	$t_{CLCL}+40$	ns
<b>Shift Register<sup>3</sup></b>							
$t_{XLXL}$	5	Serial port clock cycle time	0.75		$12t_{CLCL}$		$\mu s$
$t_{QVXH}$	5	Output data setup to clock rising edge	492		$10t_{CLCL}-133$		ns
$t_{XHGX}$	5	Output data hold after clock rising edge	80		$2t_{CLCL}-117$		ns
$t_{XHDX}$	5	Input data hold after clock rising edge	0		0		ns
$t_{XHDX}$	5	Clock rising edge to input data valid		492		$10t_{CLCL}-133$	ns
<b>External Clock</b>							
$t_{CHCX}$	6	High time	20		20	$t_{CLCL} - t_{LOW}$	ns
$t_{CLCX}$	6	Low time	20		20	$t_{CLCL} - t_{HIGH}$	ns
$t_{CLCH}$	6	Rise time		20		20	ns
$t_{CHCL}$	6	Fall time		20		20	ns

### NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- Test condition:  $T_{amb} = 0^{\circ}C$  to  $+70^{\circ}C$ ;  $V_{DD} = 5V \pm 10\%$ ;  $V_{SS} = 0V$ ; load capacitance = 80pF.

# CMOS single-chip 8-bit microcontroller with Electromagnetic Compatibility improvements

80CE654/83CE654

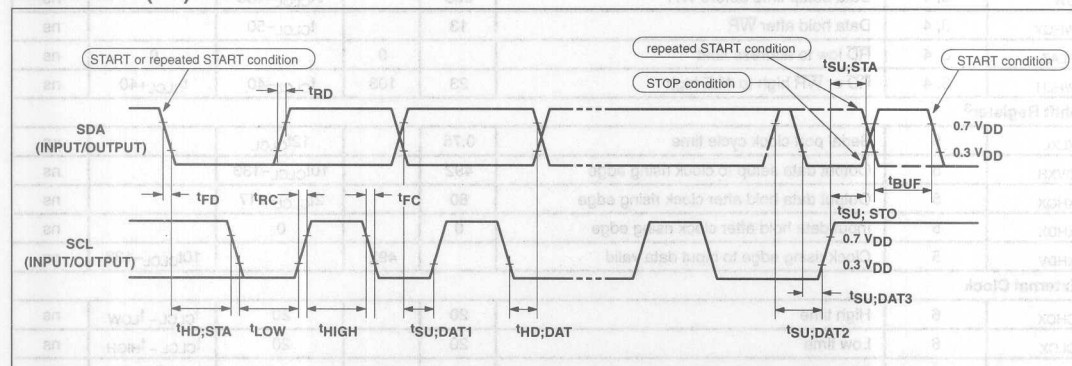
## AC ELECTRICAL CHARACTERISTICS – I<sup>2</sup>C INTERFACE

SYMBOL	PARAMETER	INPUT	OUTPUT
<b>SCL TIMING CHARACTERISTICS</b>			
$t_{HD, STA}$	START condition hold time	$\geq 14 t_{CLCL}$	$> 4.0\mu s^1$
$t_{LOW}$	SCL LOW time	$\geq 16 t_{CLCL}$	$> 4.7\mu s^1$
$t_{HIGH}$	SCL HIGH time	$\geq 14 t_{CLCL}$	$> 4.0\mu s^1$
$t_{RC}$	SCL rise time	$\leq 1\mu s$	– <sup>2</sup>
$t_{FC}$	SCL fall time	$\leq 0.3\mu s$	$< 0.3\mu s^3$
<b>SDA TIMING CHARACTERISTICS</b>			
$t_{SU, DAT1}$	Data set-up time	$\geq 250ns$	$> 20 t_{CLCL} - t_{RD}$
$t_{SU, DAT2}$	SDA set-up time (before rep. START cond.)	$\geq 250ns$	$> 1\mu s^1$
$t_{SU, DAT3}$	SDA set-up time (before STOP cond.)	$\geq 250ns$	$> 8 t_{CLCL}$
$t_{HD, DAT}$	Data hold time	$\geq 0ns$	$> 8 t_{CLCL} - t_{FC}$
$t_{SU, STA}$	Repeated START set-up time	$\geq 14 t_{CLCL}$	$> 4.7\mu s^1$
$t_{SU, STO}$	STOP condition set-up time	$\geq 14 t_{CLCL}$	$> 4.0\mu s^1$
$t_{BUF}$	Bus free time	$\geq 14 t_{CLCL}$	$> 4.7\mu s^1$
$t_{RD}$	SDA rise time	$\leq 1\mu s$	– <sup>2</sup>
$t_{FD}$	SDA fall time	$\leq 0.3\mu s$	$< 0.3\mu s^3$

### NOTES:

- At 100 kbit/s. At other bit rates this value is inversely proportional to the bit-rate of 100 kbit/s.
- Determined by the external bus-line capacitance and the external bus-line pull-resistor, this must be  $< 1\mu s$ .
- Spikes on the SDA and SCL lines with a duration of less than  $3 t_{CLCL}$  will be filtered out. Maximum capacitance on bus-lines SDA and SCL = 400pF.
- $t_{CLCL} = 1/f_{OSC}$  = one oscillator clock period at pin XTAL1. For  $62ns < t_{CLCL} < 285ns$  ( $16MHz > f_{OSC} > 3.5MHz$ ) the SIO1 interface meets the I<sup>2</sup>C-bus specification for bit-rates up to 100 kbit/s.

## TIMING SIO1 (I<sup>2</sup>C) INTERFACE



### Oscillator Circuitry

The capacitors connected to the crystal should be:  $C1 = C2 = 20pF$ .

# CMOS single-chip 8-bit microcontroller with Electromagnetic Compatibility improvements

80CE654/83CE654

## EXPLANATION OF THE AC SYMBOLS

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

- A – Address
- C – Clock
- D – Input data
- H – Logic level high
- I – Instruction (program memory contents)
- L – Logic level low, or ALE
- P – PSEN

Q – Output data

R – RD signal

t – Time

V – Valid

W – WR signal

X – No longer a valid logic level

Z – Float

Examples:  $t_{AVLL}$  = Time for address valid to ALE low.

$t_{LLPL}$  = Time for ALE low to PSEN low.

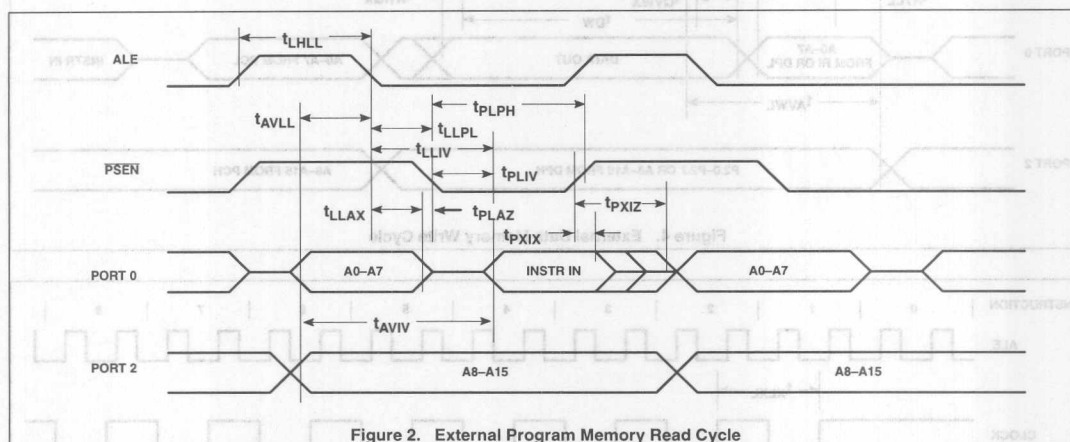


Figure 2. External Program Memory Read Cycle

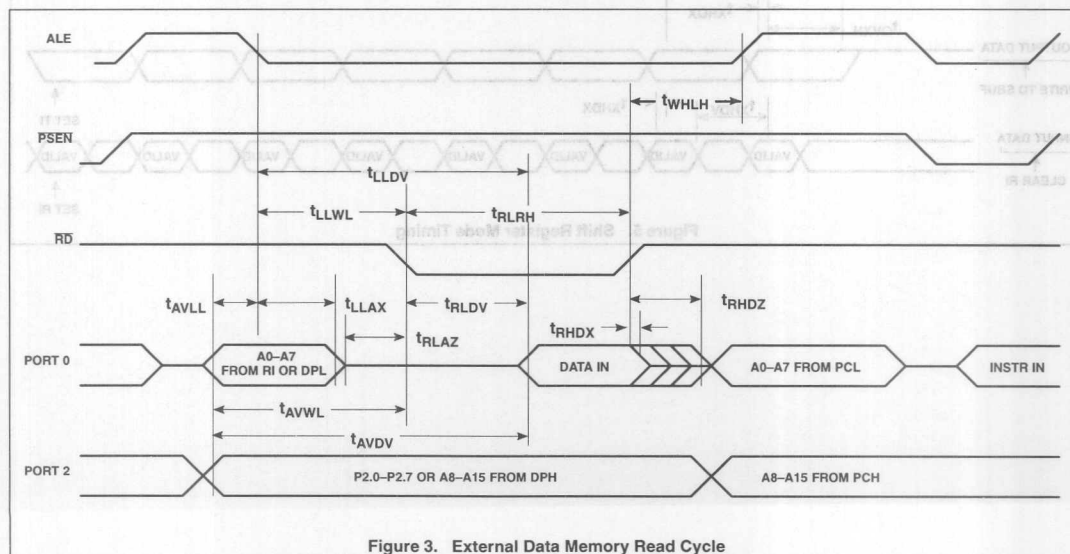
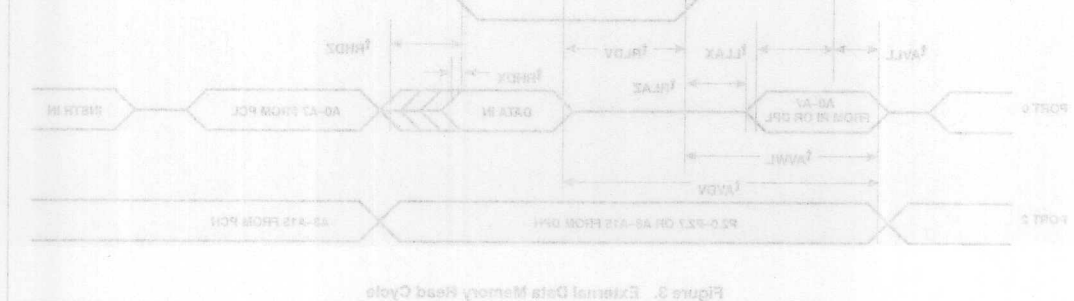
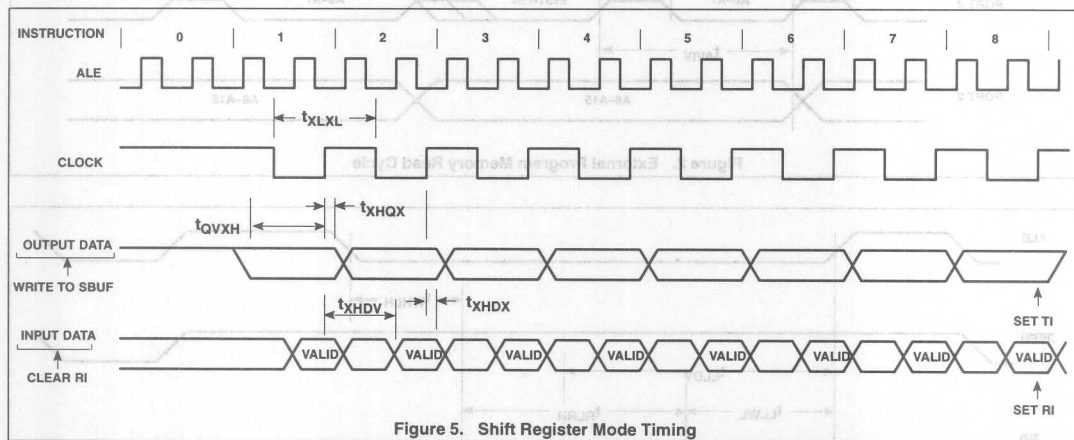
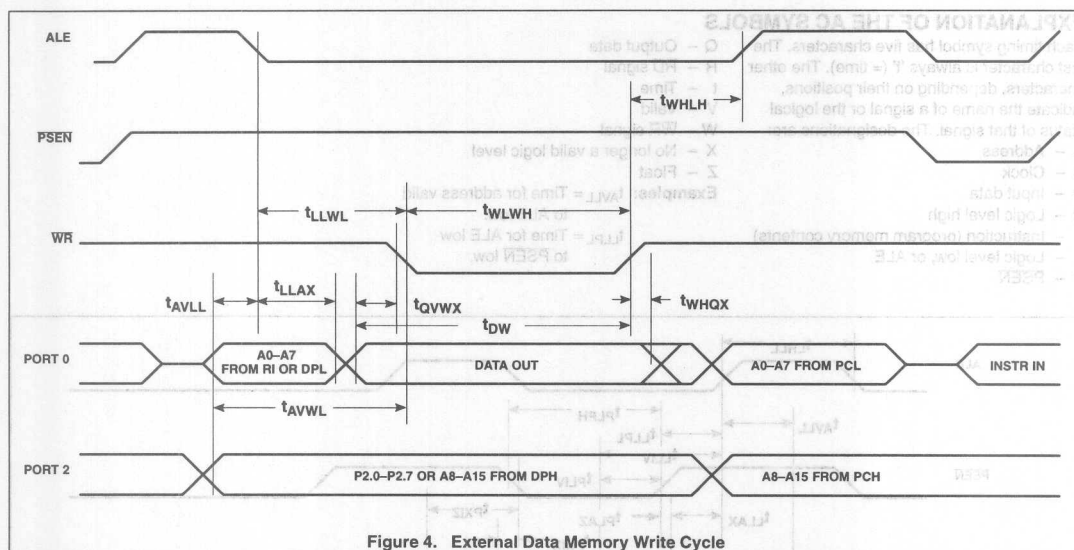


Figure 3. External Data Memory Read Cycle

# CMOS single-chip 8-bit microcontroller with Electromagnetic Compatibility improvements

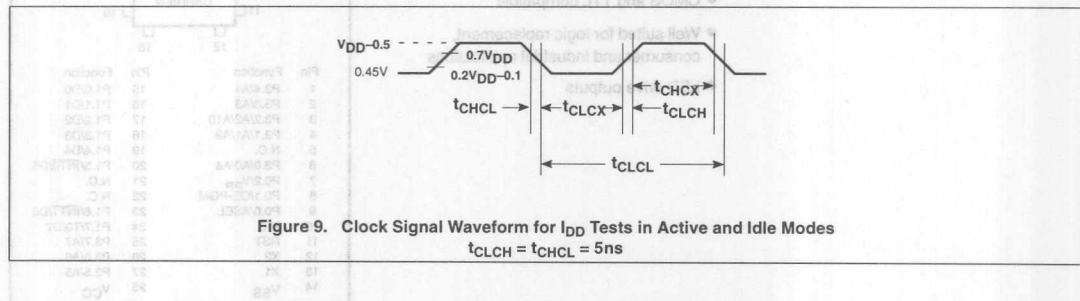
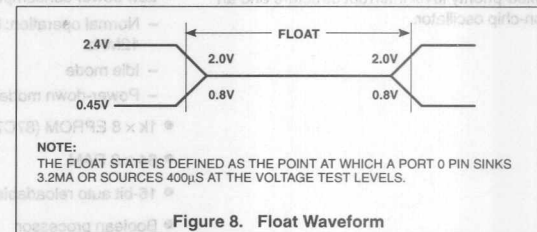
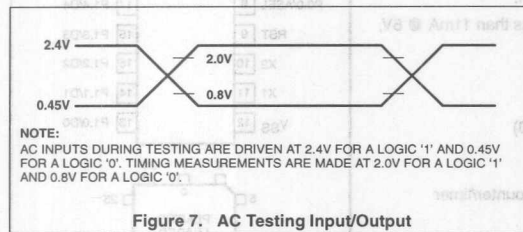
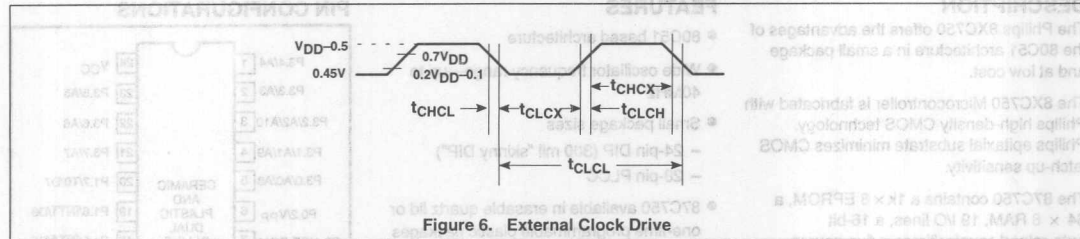
80CE654/83CE654





# CMOS single-chip 8-bit microcontroller with Electromagnetic Compatibility improvements

80CE654/83CE654



DRAWING NUMBER	FREQUENCY	TEMPERATURE RANGE °C AND PACKAGE	EPROM	ROM
0088B	0.5 to 10MHz	0 to +70, Ceramic Dual In-line Package	UV	P8C750BEP FA
0088B	0.5 to 10MHz	-40 to +65, Ceramic Dual In-line Package	UV	P8C750BEP FA
0410D	0.5 to 10MHz	0 to +70, Plastic Dual In-line Package	OTP	P8C750BEP N
0410D	0.5 to 10MHz	-40 to +65, Plastic Dual In-line Package	OTP	P8C750BEP N
0407F	0.5 to 10MHz	0 to +70, Plastic Lead Chip Carrier	OTP	P8C750BEP A
0407F	0.5 to 10MHz	-40 to +65, Plastic Lead Chip Carrier	OTP	P8C750BEP A
0410D	0.5 to 10MHz	0 to +70, Plastic Dual In-line Package	OTP	P8C750BEP N
0410D	0.5 to 10MHz	-40 to +65, Plastic Dual In-line Package	OTP	P8C750BEP N
0407F	0.5 to 10MHz	0 to +70, Plastic Lead Chip Carrier	OTP	P8C750BEP A
0407F	0.5 to 10MHz	-40 to +65, Plastic Lead Chip Carrier	OTP	P8C750BEP A
0410D	0.5 to 10MHz	0 to +70, Ceramic Dual In-line Package	UV	P8C750BEP FA
0410D	0.5 to 10MHz	-40 to +65, Ceramic Dual In-line Package	UV	P8C750BEP FA



Purchase of Philips' I<sup>2</sup>C components conveys a license under the Philips' I<sup>2</sup>C patent to use the components in the I<sup>2</sup>C-system provided the system conforms to the I<sup>2</sup>C specifications defined by Philips.

## CMOS single-chip 8-bit microcontrollers

83C750/87C750

## DESCRIPTION

The Philips 8XC750 offers the advantages of the 80C51 architecture in a small package and at low cost.

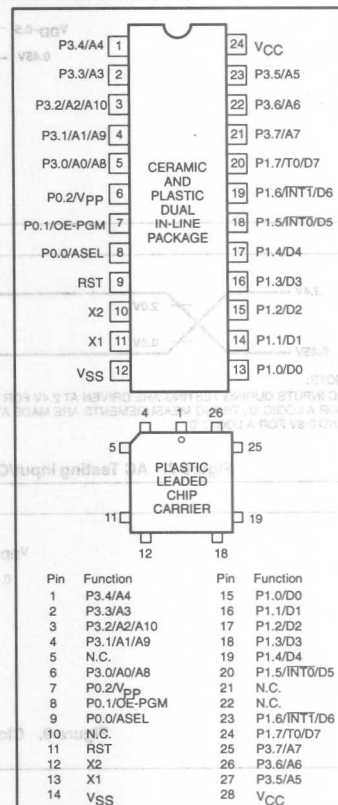
The 8XC750 Microcontroller is fabricated with Philips high-density CMOS technology. Philips epitaxial substrate minimizes CMOS latch-up sensitivity.

The 87C750 contains a 1k x 8 EPROM, a 64 x 8 RAM, 19 I/O lines, a 16-bit auto-reload counter/timer, a five-source, fixed-priority level interrupt structure and an on-chip oscillator.

## FEATURES

- 80C51 based architecture
- Wide oscillator frequency range—up to 40MHz
- Small package sizes
  - 24-pin DIP (300 mil "skinny DIP")
  - 28-pin PLCC
- 87C750 available in erasable quartz lid or one-time programmable plastic packages
- Low power consumption:
  - Normal operation: less than 11mA @ 5V, 12MHz
  - Idle mode
  - Power-down mode
- 1k x 8 EPROM (87C750)
- 64 x 8 RAM
- 16-bit auto reloadable counter/timer
- Boolean processor
- CMOS and TTL compatible
- Well suited for logic replacement, consumer and industrial applications
- LED drive outputs

## PIN CONFIGURATIONS



## ORDERING INFORMATION

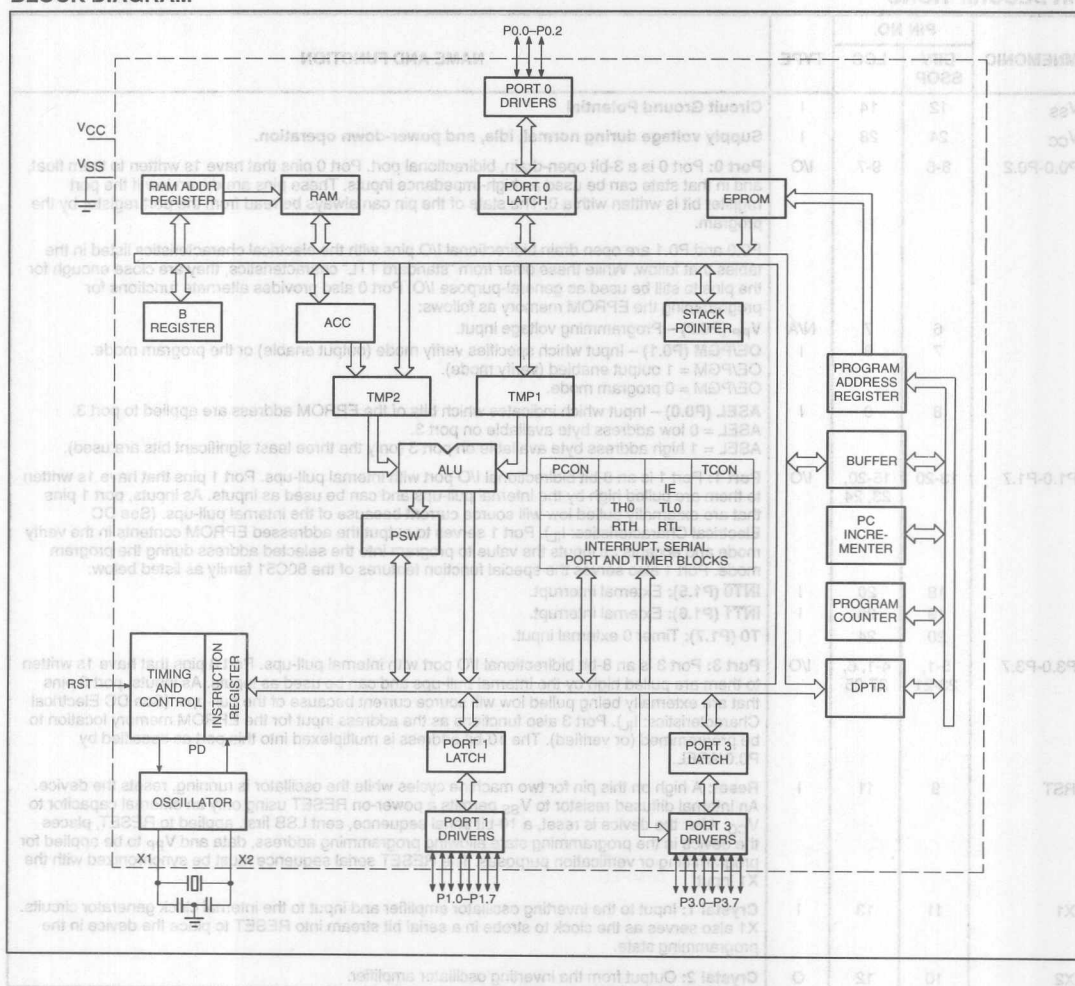
ROM	EPROM <sup>1</sup>	TEMPERATURE RANGE °C AND PACKAGE	FREQUENCY	DRAWING NUMBER
	P87C750EBF FA UV	0 to +70, Ceramic Dual In-line Package	3.5 to 16MHz	0586B
	P87C750EFF FA UV	-40 to +85, Ceramic Dual In-line Package	3.5 to 16MHz	0586B
P83C750EBP N	P87C750EBP N OTP	0 to +70, Plastic Dual In-line Package	3.5 to 16MHz	0410D
P83C750EFP N	P87C750EFP N OTP	-40 to +85, Plastic Dual In-line Package	3.5 to 16MHz	0410D
P83C750EBA A	P87C750EBA A OTP	0 to +70, Plastic Lead Chip Carrier	3.5 to 16MHz	0401F
P83C750EFA A	P87C750EFA A OTP	-40 to +85, Plastic Lead Chip Carrier	3.5 to 16MHz	0401F
P83C750PBP N	P87C750PBP N OTP	0 to +70, Plastic Dual In-line Package	3.5 to 40MHz	0410D
P83C750PFP N	P87C750PFP N OTP	-40 to +85, Plastic Dual In-line Package	3.5 to 40MHz	0410D
P83C750PBA A	P87C750PBA A OTP	0 to +70, Plastic Lead Chip Carrier	3.5 to 40MHz	0401F
P83C750PFA A	P87C750PFA A OTP	-40 to +85, Plastic Lead Chip Carrier	3.5 to 40MHz	0401F
	P87C750PBF FA UV	0 to +70, Ceramic Dual In-line Package	3.5 to 40MHz	0586B
	P87C750PFF FA UV	-40 to +85, Ceramic Dual In-line Package	3.5 to 40MHz	0586B

## NOTE:

1. OTP = One Time Programmable EPROM. UV = UV Erasable EPROM.

## CMOS single-chip 8-bit microcontrollers 83C750/87C750

### BLOCK DIAGRAM



## CMOS single-chip 8-bit microcontrollers 83C750/87C750

## PIN DESCRIPTIONS

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP/SSOP	LCC		
V <sub>SS</sub>	12	14	I	<b>Circuit Ground Potential</b>
V <sub>CC</sub>	24	28	I	<b>Supply voltage during normal, idle, and power-down operation.</b>
P0.0-P0.2	8-6	9-7	I/O	<p><b>Port 0:</b> Port 0 is a 3-bit open-drain, bidirectional port. Port 0 pins that have 1s written to them float, and in that state can be used as high-impedance inputs. These pins are driven low if the port register bit is written with a 0. The state of the pin can always be read from the port register by the program.</p> <p>P0.0 and P0.1 are open drain bidirectional I/O pins with the electrical characteristics listed in the tables that follow. While these differ from "standard TTL" characteristics, they are close enough for the pins to still be used as general-purpose I/O. Port 0 also provides alternate functions for programming the EPROM memory as follows:</p> <p><b>V<sub>PP</sub> (P0.2)</b> – Programming voltage input.</p> <p><b>OE/PGM (P0.1)</b> – Input which specifies verify mode (output enable) or the program mode. OE/PGM = 1 output enabled (verify mode). OE/PGM = 0 program mode.</p> <p><b>ASEL (P0.0)</b> – Input which indicates which bits of the EPROM address are applied to port 3. ASEL = 0 low address byte available on port 3. ASEL = 1 high address byte available on port 3 (only the three least significant bits are used).</p>
P1.0-P1.7	6	7	N/A	
	7	8	I	
	8	9	I	
	13-20	15-20, 23, 24	I/O	<p><b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I<sub>IL</sub>). Port 1 serves to output the addressed EPROM contents in the verify mode and accepts as inputs the value to program into the selected address during the program mode. Port 1 also serves the special function features of the 80C51 family as listed below:</p> <p><b>INT0 (P1.5):</b> External interrupt.</p> <p><b>INT1 (P1.6):</b> External interrupt.</p> <p><b>T0 (P1.7):</b> Timer 0 external input.</p>
P3.0-P3.7	5-1, 23-21	4-1, 6, 27-25	I/O	<p><b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: I<sub>IL</sub>). Port 3 also functions as the address input for the EPROM memory location to be programmed (or verified). The 10-bit address is multiplexed into this port as specified by P0.0/ASEL.</p>
RST	9	11	I	<p><b>Reset:</b> A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to V<sub>SS</sub> permits a power-on RESET using only an external capacitor to V<sub>CC</sub>. After the device is reset, a 10-bit serial sequence, sent LSB first, applied to RESET, places the device in the programming state allowing programming address, data and V<sub>PP</sub> to be applied for programming or verification purposes. The RESET serial sequence must be synchronized with the X1 input.</p>
X1	11	13	I	<p><b>Crystal 1:</b> Input to the inverting oscillator amplifier and input to the internal clock generator circuits. X1 also serves as the clock to strobe in a serial bit stream into RESET to place the device in the programming state.</p>
X2	10	12	O	<p><b>Crystal 2:</b> Output from the inverting oscillator amplifier.</p>

## CMOS single-chip 8-bit microcontrollers 83C750/87C750

## OSCILLATOR

## CHARACTERISTICS

X1 and X2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator.

To drive the device from an external clock source, X1 should be driven while X2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

## RESET

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. To insure a good power-up reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few milliseconds) plus two machine cycles. At power-up, the voltage on  $V_{CC}$  and RST must come up at the same time for a proper start-up.

## IDLE MODE

In idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

## POWER-DOWN MODE

In the power-down mode, the oscillator is stopped and the instruction to invoke power-down is the last instruction executed. Only the contents of the on-chip RAM are preserved. A hardware reset is the only way to terminate the power-down mode. The control bits for the reduced power modes are in the special function register PCON.

Table 1. External Pin Status During Idle and Power-Down Modes

MODE	Port 0	Port 1	Port 2
Idle	Data	Data	Data
Power-down	Data	Data	Data

## DIFFERENCES BETWEEN THE 8XC750 AND THE 80C51

## Program Memory

On the 8XC750, program memory is 1024 bytes long and is not externally expandable, so the 80C51 instructions MOVX, LJMP, and LCALL are not implemented. The only fixed locations in program memory are the addresses at which execution is taken up in response to reset and interrupts, which are as follows:

Event	Program Memory Address
Reset	000
External INT0	003
Counter/timer 0	00B
External INT1	013

## Counter/Timer Subsystem

The 87C750 has one counter/timer called timer/counter 0. Its operation is similar to mode 2 operation on the 80C51, but is extended to 16 bits with 16 bits of autoloader. The controls for this counter are centralized in a single register called TCON.

## Interrupt Subsystem – Fixed Priority

The IP register and the 2-level interrupt system of the 80C51 are eliminated. Simultaneous interrupt conditions are resolved by a single-level, fixed priority as follows:

Highest priority: Pin INT0  
Counter/timer flag 0  
Pin INT1

## Special Function Register Addresses

Special function registers for the 8XC750 are identical to those of the 80C51, except for the changes listed below:

80C51 special function registers not present in the 8XC750 are TMOD (89), P2 (A0) and IP (B8). The 80C51 registers TH1 and TL1 are replaced with the 87C750 registers RTH and RTL respectively (refer to Table 2).

89H	Timer mode control	TMOD
8AH	Timer low byte	TL0
8BH	Timer high byte	TH0
8CH	Timer low reload	RL0
8DH	Timer high reload	RH0

UNIT	RATING	PARAMETER
°C	-55 to +150	Storage temperature range
V	-0.5 to +5.5	Voltage from $V_{CC}$ to $V_{SS}$
V	-0.5 to $V_{CC} + 0.5$	Voltage from any pin to $V_{SS}$ (except $V_{PP}$ )
W	1.0	Power dissipation
V	0 to +12.0	Voltage on $V_{PP}$ pin to $V_{SS}$
Am	10	Maximum IOL per I/O pin

## NOTES:

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section is not implied.
- This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nevertheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maximum charge. Nevertheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maximum charge.



**Table 2. 87C750 Special Function Registers**

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB				LSB				
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
DPTR:	Data pointer (2 bytes)										
DPH	High byte	83H									00H
DPL	Low byte	82H									00H
IE*#	Interrupt enable	A8H	AF	AE	AD	AC	AB	AA	A9	A8	00H
			EA	—	—	—	—	EX1	ET0	EX0	00H
P0*#	Port 0	80H									xxxxx111B
P1*	Port 1	90H	97	96	95	94	93	92	91	90	FFH
P3*	Port 3	B0H	T0	INT1	INT0	—	—	—	—	—	FFH
			B7	B6	B5	B4	B3	B2	B1	B0	FFH
PCON#	Power control	87H							PD	IDL	xxxxxx00B
PSW*	Program status word	D0H	D7	D6	D5	D4	D3	D2	D1	D0	00H
			CY	AC	F0	RS1	RS0	OV	—	P	00H
SP	Stack pointer	81H									07H
			8F	8E	8D	8C	8B	8A	89	88	00H
TCON*#	Timer/counter control	88H	GATE	C/T	TF	TR	IE0	IT0	IE1	IT1	00H
TL#	Timer low byte	8AH									00H
TH#	Timer high byte	8CH									00H
RTL#	Timer low reload	8BH									00H
RTH#	Timer high reload	8DH									00H

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.

## ABSOLUTE MAXIMUM RATINGS<sup>1, 2</sup>

PARAMETER	RATING	UNIT
Storage temperature range	–65 to +150	°C
Voltage from $V_{CC}$ to $V_{SS}$	–0.5 to +6.5	V
Voltage from any pin to $V_{SS}$ (except $V_{PP}$ )	–0.5 to $V_{CC} + 0.5$	V
Power dissipation	1.0	W
Voltage on $V_{PP}$ pin to $V_{SS}$	0 to +13.0	V
Maximum $I_{OL}$ per I/O pin	10	mA

### NOTES:

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
- This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.

## CMOS single-chip 8-bit microcontrollers 83C750/87C750

## DC ELECTRICAL CHARACTERISTICS

 $T_{amb} = 0^{\circ}\text{C to } +70^{\circ}\text{C or } -40^{\circ}\text{C to } +85^{\circ}\text{C}, V_{CC} = 5\text{V} \pm 10\%, V_{SS} = 0\text{V}^1$ 

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			MIN	MAX	
$V_{IL}$	Input low voltage		-0.5	$0.2V_{DD} - 0.1$	V
$V_{IH}$	Input high voltage, except X1, RST		$0.2V_{CC} + 0.9$	$V_{CC} + 0.5$	V
$V_{IH1}$	Input high voltage, X1, RST		$0.7V_{CC}$	$V_{CC} + 0.5$	V
$V_{OL}$	Output low voltage, ports 1 and 3	$I_{OL} = 1.6\text{mA}^2$		0.45	V
$V_{OL1}$	Output low voltage, port 0	$I_{OL} = 3.2\text{mA}^2$		0.45	V
$V_{OH}$	Output high voltage, ports 1 and 3	$I_{OH} = -60\mu\text{A}$	2.4		V
		$I_{OH} = -25\mu\text{A}$	$0.75V_{CC}$		V
		$I_{OH} = -10\mu\text{A}$	$0.9V_{CC}$		V
C	Capacitance			10	pF
$I_{IL}$	Logical 0 input current, ports 1 and 3	$V_{IN} = 0.45\text{V}$		-50	$\mu\text{A}$
$I_{TL}$	Logical 1 to 0 transition current, ports 1 and 3 <sup>3</sup>	$V_{IN} = 2\text{V}$ (0 to $+70^{\circ}\text{C}$ )		-650	$\mu\text{A}$
		$V_{IN} = 2\text{V}$ ( $-40$ to $+85^{\circ}\text{C}$ )		-750	$\mu\text{A}$
$I_{LI}$	Input leakage current, port 0	$0.45 < V_{IN} < V_{CC}$		$\pm 10$	$\mu\text{A}$
$R_{RST}$	Internal pull-down resistor		25	175	k $\Omega$
$C_{IO}$	Pin capacitance	Test freq = 1MHz, $T_{amb} = 25^{\circ}\text{C}$		10	pF
$I_{PD}$	Power-down current <sup>4</sup>	$V_{CC} = 2$ to $V_{CC}$ max		50	$\mu\text{A}$
$V_{PP}$	$V_{PP}$ program voltage	$V_{SS} = 0\text{V}$ $V_{CC} = 5\text{V} \pm 10\%$ $T_{amb} = 21^{\circ}\text{C to } 27^{\circ}\text{C}$	12.5	13.0	V
$I_{PP}$	Program current	$V_{PP} = 13.0\text{V}$		50	mA
$I_{CC}$	Supply current (see Figure 2) <sup>5, 6</sup>				

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to  $V_{SS}$  unless otherwise noted.
- Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:  
 Maximum  $I_{OL}$  per port pin: 10mA (NOTE: This is  $85^{\circ}\text{C}$  spec.)  
 Maximum  $I_{OL}$  per 8-bit port: 26mA  
 Maximum total  $I_{OL}$  for all outputs: 67mA  
 If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
- Pins of ports 1 and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when  $V_{IN}$  is approximately 2V.
- Power-down  $I_{CC}$  is measured with all output pins disconnected; port 0 =  $V_{CC}$ ; X2, X1 n.c.; RST =  $V_{SS}$ .
- Active  $I_{CC}$  is measured with all output pins disconnected; X1 driven with  $t_{CLCH}$ ,  $t_{CHCL} = 5\text{ns}$ ,  $V_{IL} = V_{SS} + 0.5\text{V}$ ,  $V_{IH} = V_{CC} - 0.5\text{V}$ ; X2 n.c.; RST = port 0 =  $V_{CC}$ .  $I_{CC}$  will be slightly higher if a crystal oscillator is used.
- Idle  $I_{CC}$  is measured with all output pins disconnected; X1 driven with  $t_{CLCH}$ ,  $t_{CHCL} = 5\text{ns}$ ,  $V_{IL} = V_{SS} + 0.5\text{V}$ ,  $V_{IH} = V_{CC} - 0.5\text{V}$ ; X2 n.c.; port 0 =  $V_{CC}$ ; RST =  $V_{SS}$ .

## AC ELECTRICAL CHARACTERISTICS

 $T_{amb} = 0^{\circ}\text{C to } +70^{\circ}\text{C or } -40^{\circ}\text{C to } +85^{\circ}\text{C}, V_{CC} = 5\text{V} \pm 10\%, V_{SS} = 0\text{V}^1, 2$ 

SYMBOL	PARAMETER	VARIABLE CLOCK				UNIT
		MIN	MAX	MIN	MAX	
$1/t_{CLCL}$	Oscillator frequency:	3.5	16	3.5	40	MHz
External Clock (Figure 1)						
$t_{CHCX}$	High time	20		10		ns
$t_{CLCX}$	Low time	20		10		ns
$t_{CLCH}$	Rise time		20		20	ns
$t_{CHCL}$	Fall time		20		20	ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to  $V_{SS}$  unless otherwise noted.
- Load capacitance for ports = 80pF.

## CMOS single-chip 8-bit microcontrollers 83C750/87C750

## EXPLANATION OF THE AC SYMBOLS

In defining the clock waveform, care must be taken not to exceed the MIN or MAX limits of the AC electrical characteristics table. Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

C — Clock  
D — Input data

H — Logic level high  
L — Logic level low  
Q — Output data  
T — Time  
V — Valid  
X — No longer a valid logic level  
Z — Float

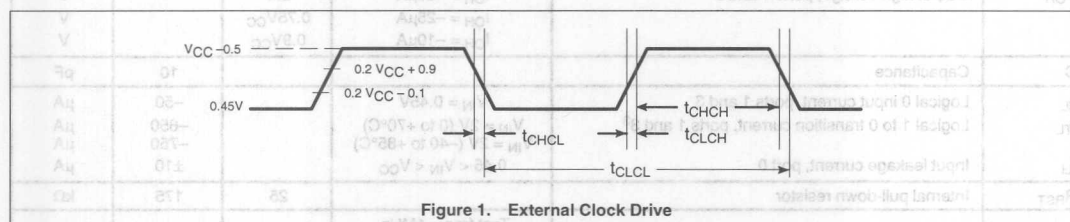
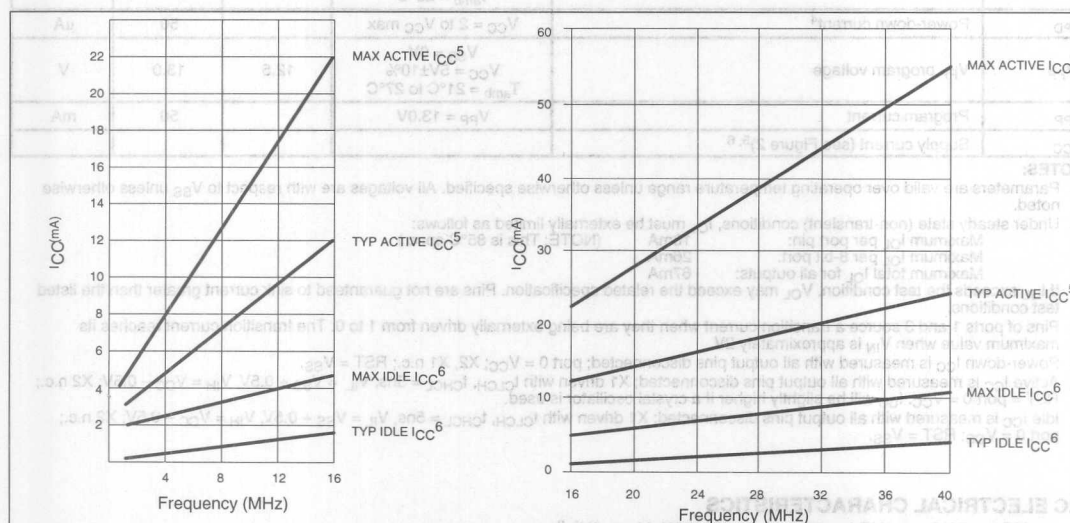


Figure 1. External Clock Drive

Figure 2.  $I_{CC}$  vs. Frequency

Maximum  $I_{CC}$  values taken at  $V_{CC}$  max and worst case temperature.

Typical  $I_{CC}$  values taken at  $V_{CC} = 5.0V$  and  $25^{\circ}C$ .

Notes 5 and 6 refer to DC Electrical Characteristics.

## CMOS single-chip 8-bit microcontrollers

## 83C750/87C750

**87C750 PROGRAMMING CONSIDERATIONS****EPROM Characteristics**

The 87C750 is programmed by using a modified Quick-Pulse Programming algorithm similar to that used for devices such as the 87C451 and 87C51. It differs from these devices in that a serial data stream is used to place the 87C750 in the programming mode.

Figure 3 shows a block diagram of the programming configuration for the 87C750. Port pin P0.2 is used as the programming voltage supply input ( $V_{PP}$  signal). Port pin P0.1 is used as the program (PGM) signal. This pin is used for the 25 programming pulses.

Port 3 is used as the address input for the byte to be programmed and accepts both the high and low components of the eleven bit address. Multiplexing of these address components is performed using the ASEL input. The user should drive the ASEL input high and then drive port 3 with the high order bits of the address. ASEL should remain high for at least 13 clock cycles. ASEL may then be driven low which latches the high order bits of the address internally. The high address should remain on port 3 for at least two clock cycles after ASEL is driven low. Port 3 may then be driven with the low byte of the address. The low address will be internally stable 13 clock cycles later. The address will remain stable provided that the low byte placed on port 3 is held stable and ASEL is kept low. **Note:** ASEL needs to be pulsed high only to change the high byte of the address.

Port 1 is used as a bidirectional data bus during programming and verify operations. During programming mode, it accepts the byte to be programmed. During verify mode, it provides the contents of the EPROM location specified by the address which has been supplied to Port 3.

The XTAL1 pin is the oscillator input and receives the master system clock. This clock should be between 1.2 and 6MHz.

The RESET pin is used to accept the serial data stream that places the 87C750 into various programming modes. This pattern consists of a 10-bit code with the LSB sent first. Each bit is synchronized to the clock input, X1.

**Programming Operation**

Figures 4 and 5 show the timing diagrams for the program/verify cycle. RESET should

initially be held high for at least two machine cycles. P0.1 (PGM) and P0.2 ( $V_{PP}$ ) will be at  $V_{OH}$  as a result of the RESET operation. At this point, these pins function as normal quasi-bidirectional I/O ports and the programming equipment may pull these lines low. However, prior to sending the 10-bit code on the RESET pin, the programming equipment should drive these pins high ( $V_{IH}$ ). The RESET pin may now be used as the serial data input for the data stream which places the 87C750 in the programming mode. Data bits are sampled during the clock high time and thus should only change during the time that the clock is low. Following transmission of the last data bit, the RESET pin should be held low.

Next the address information for the location to be programmed is placed on port 3 and ASEL is used to perform the address multiplexing, as previously described. At this time, port 1 functions as an output.

A high voltage  $V_{PP}$  level is then applied to the  $V_{PP}$  input (P0.2). (This sets Port 1 as an input port). The data to be programmed into the EPROM array is then placed on Port 1. This is followed by a series of programming pulses applied to the PGM/ pin (P0.1). These pulses are created by driving P0.1 low and then high. This pulse is repeated until a total of 25 programming pulses have occurred. At the conclusion of the last pulse, the PGM/ signal should remain high.

The  $V_{PP}$  signal may now be driven to the  $V_{OH}$  level, placing the 87C750 in the verify mode. (Port 1 is now used as an output port). After four machine cycles (48 clock periods), the contents of the addressed location in the EPROM array will appear on Port 1.

The next programming cycle may now be initiated by placing the address information at the inputs of the multiplexed buffers, driving the  $V_{PP}$  pin to the  $V_{PP}$  voltage level, providing the byte to be programmed to Port1 and issuing the 26 programming pulses on the PGM/ pin, bringing  $V_{PP}$  back down to the  $V_C$  level and verifying the byte.

**Programming Modes**

The 87C750 has four programming features incorporated within its EPROM array. These include the USER EPROM for storage of the application's code, a 16-byte encryption key array and two security bits. Programming and verification of these four elements are selected by a combination of the serial data stream applied to the RESET pin and the voltage levels applied to port pins P0.1 and

P0.2. The various combinations are shown in Table 3.

**Encryption Key Table**

The 87C750 includes a 16-byte EPROM array that is programmable by the end user. The contents of this array can then be used to encrypt the program memory contents during a program memory verify operation. When a program memory verify operation is performed, the contents of the program memory location is XNOR'ed with one of the bytes in the 16-byte encryption table. The resulting data pattern is then provided to port 1 as the verify data. The encryption mechanism can be disable, in essence, by leaving the bytes in the encryption table in their erased state (FFH) since the XNOR product of a bit with a logical one will result in the original bit. The encryption bytes are mapped with the code memory in 16-byte groups. The first byte in code memory will be encrypted with the first byte in the encryption table; the second byte in code memory will be encrypted with the second byte in the encryption table and so forth up to and including the 16th byte. The encryption repeats in 16-byte groups; the 17th byte in the code memory will be encrypted with the first byte in the encryption table, and so forth.

**Security Bits**

Two security bits, security bit 1 and security bit 2, are provided to limit access to the USER EPROM and encryption key arrays. Security bit 1 is the program inhibit bit, and once programmed performs the following functions:

1. Additional programming of the USER EPROM is inhibited.
2. Additional programming of the encryption key is inhibited.
3. Verification of the encryption key is inhibited.
4. Verification of the USER EPROM and the security bit levels may still be performed.

(If the encryption key array is being used, this security bit should be programmed by the user to prevent unauthorized parties from reprogramming the encryption key to all logical zero bits. Such programming would provide data during a verify cycle that is the logical complement of the USER EPROM contents).

Security bit 2, the verify inhibit bit, prevents verification of both the USER EPROM array and the encryption key arrays. The security bit levels may still be verified.



## CMOS single-chip 8-bit microcontrollers 83C750/87C750

### Programming and Verifying Security Bits

Security bits are programmed employing the same techniques used to program the USER EPROM and KEY arrays using serial data streams and logic levels on port pins indicated in Table 3. When programming either security bit, it is not necessary to provide address or data information to the 87C750 on ports 1 and 3.

Verification occurs in a similar manner using the RESET serial stream shown in Table 3. Port 3 is not required to be driven and the results of the verify operation will appear on ports 1.6 and 1.7.

Ports 1.7 contains the security bit 1 data and is a logical one if programmed and a logical zero if erased. Likewise, P1.6 contains the security bit 2 data and is a logical one if programmed and a logical zero if erased.

### Erasure Characteristics

Erasure of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent

erasure. For this and secondary effects, it is recommended that an opaque label be placed over the window. For elevated temperature or environments where solvents are being used, apply Kapton tape Flourless part number 2345-5 or equivalent.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 angstroms) to an integrated dose of at least 15W-s/cm<sup>2</sup>. Exposing the EPROM to an ultraviolet lamp of 12,000µW/cm<sup>2</sup> rating for 20 to 39 minutes, at a distance of about 1 inch, should be sufficient.

Erasure leaves the array in an all 1s state.

Table 3. Implementing Program/Verify Modes

OPERATION	SERIAL CODE	P0.1 (PGM)	P0.2 (Vpp)
Program user EPROM	296H	1	V <sub>PP</sub>
Verify user EPROM	296H	V <sub>IH</sub>	V <sub>IH</sub>
Program key EPROM	292H	1	V <sub>PP</sub>
Verify key EPROM	292H	V <sub>IH</sub>	V <sub>IH</sub>
Program security bit 1	29AH	1	V <sub>PP</sub>
Program security bit 2	298H	1	V <sub>PP</sub>
Verify security bits	29AH	V <sub>IH</sub>	V <sub>IH</sub>

### NOTE:

1. Pulsed from V<sub>IH</sub> to V<sub>IL</sub> and returned to V<sub>IH</sub>.

### EPROM PROGRAMMING AND VERIFICATION

T<sub>amb</sub> = 21°C to +27°C, V<sub>CC</sub> = 5V ±10%, V<sub>SS</sub> = 0V

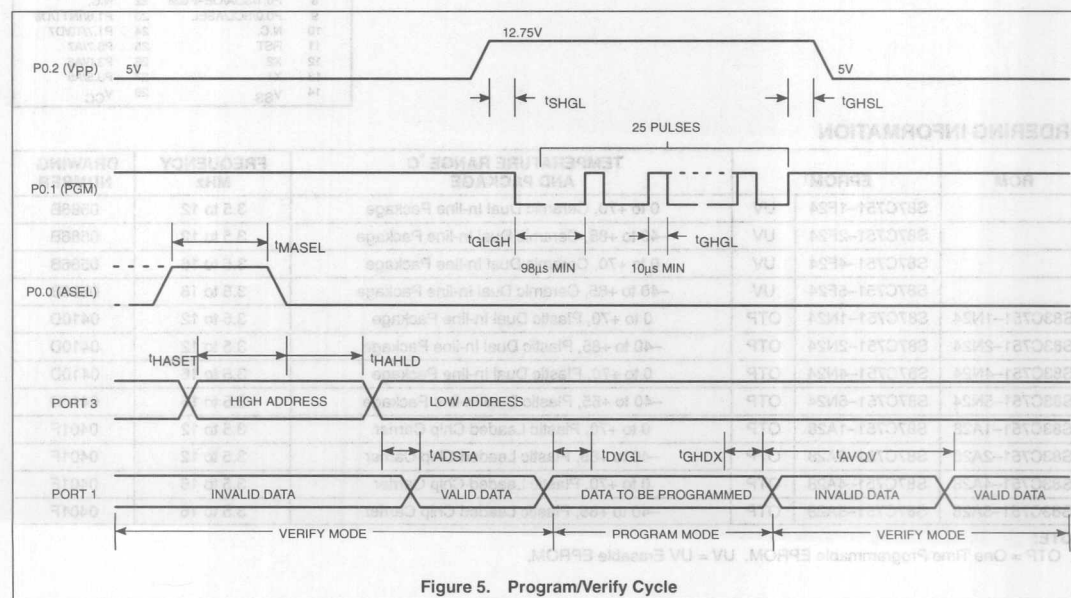
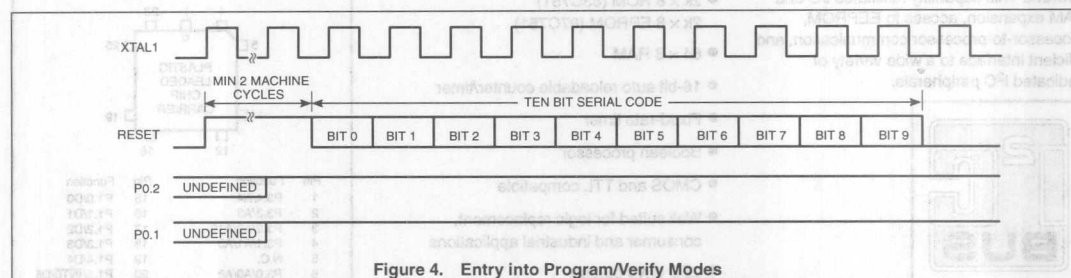
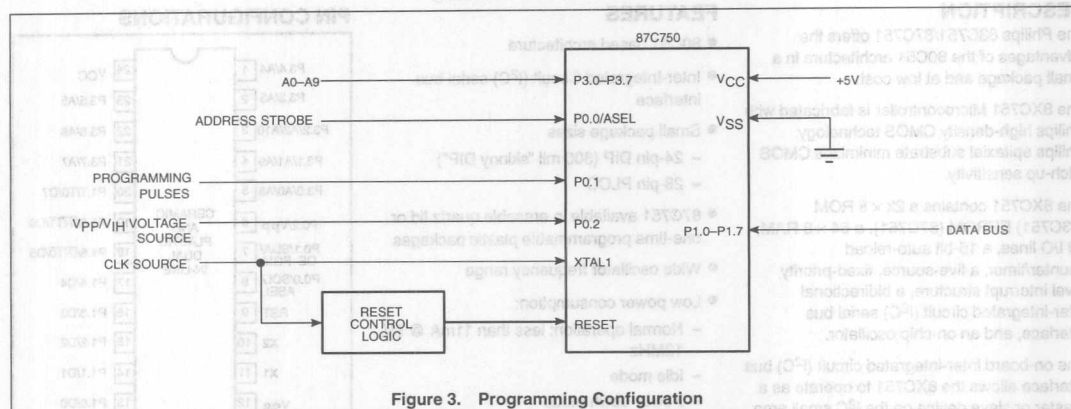
SYMBOL	PARAMETER	MIN	MAX	UNIT
1/t <sub>CLCL</sub>	Oscillator/clock frequency	1.2	6	MHz
t <sub>AVGL</sub> <sup>1</sup>	Address setup to P0.1 (PROG-) low	10µs + 24t <sub>CLCL</sub>		
t <sub>GHAX</sub>	Address hold after P0.1 (PROG-) high	48t <sub>CLCL</sub>		
t <sub>DVGL</sub>	Data setup to P0.1 (PROG-) low	38t <sub>CLCL</sub>		
t <sub>DVGL</sub>	Data setup to P0.1 (PROG-) low	38t <sub>CLCL</sub>		
t <sub>GHDX</sub>	Data hold after P0.1 (PROG-) high	36t <sub>CLCL</sub>		
t <sub>SHGL</sub>	V <sub>PP</sub> setup to P0.1 (PROG-) low	10		µs
t <sub>GHSL</sub>	V <sub>PP</sub> hold after P0.1 (PROG-)	10		µs
t <sub>GLGH</sub>	P0.1 (PROG-) width	90	110	µs
t <sub>AVQV</sub> <sup>2</sup>	V <sub>PP</sub> low (V <sub>CC</sub> ) to data valid		48t <sub>CLCL</sub>	
t <sub>GHGL</sub>	P0.1 (PROG-) high to P0.1 (PROG-) low	10		µs
t <sub>SYNL</sub>	P0.0 (sync pulse) low	4t <sub>CLCL</sub>		
t <sub>SYNH</sub>	P0.0 (sync pulse) high	8t <sub>CLCL</sub>		
t <sub>MASEL</sub>	ASEL high time	13t <sub>CLCL</sub>		
t <sub>MAHLD</sub>	Address hold time	2t <sub>CLCL</sub>		
t <sub>HASET</sub>	Address setup to ASEL	13t <sub>CLCL</sub>		
t <sub>ADSTA</sub>	Low address to valid data		48t <sub>CLCL</sub>	

### NOTES:

- Address should be valid at least 24t<sub>CLCL</sub> before the rising edge of P0.2 (V<sub>PP</sub>).
- For a pure verify mode, i.e., no program mode in between, t<sub>AVQV</sub> is 14t<sub>CLCL</sub> maximum.



## CMOS single-chip 8-bit microcontrollers 87C750/87C750



# CMOS single-chip 8-bit microcontroller 83C751/87C751

## DESCRIPTION

The Philips 83C751/87C751 offers the advantages of the 80C51 architecture in a small package and at low cost.

The 83C751 Microcontroller is fabricated with Philips high-density CMOS technology. Philips epitaxial substrate minimizes CMOS latch-up sensitivity.

The 83C751 contains a  $2k \times 8$  ROM (83C751) EPROM (87C751), a  $64 \times 8$  RAM, 19 I/O lines, a 16-bit auto-reload counter/timer, a five-source, fixed-priority level interrupt structure, a bidirectional inter-integrated circuit (I<sup>2</sup>C) serial bus interface, and an on-chip oscillator.

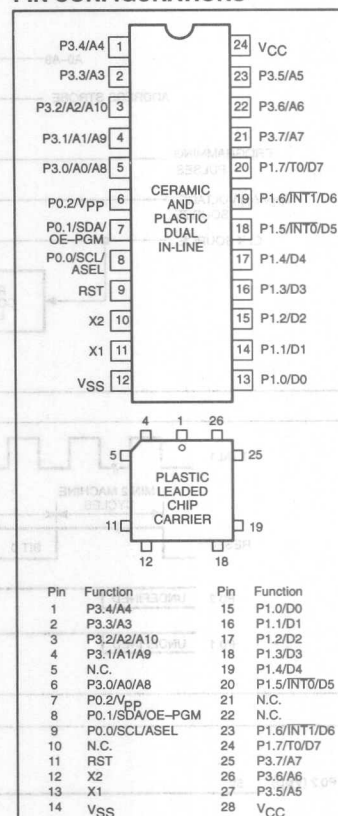
The on-board inter-integrated circuit (I<sup>2</sup>C) bus interface allows the 83C751 to operate as a master or slave device on the I<sup>2</sup>C small area network. This capability facilitates I/O and RAM expansion, access to EEPROM, processor-to-processor communication, and efficient interface to a wide variety of dedicated I<sup>2</sup>C peripherals.



## FEATURES

- 80C51 based architecture
- Inter-Integrated Circuit (I<sup>2</sup>C) serial bus interface
- Small package sizes
  - 24-pin DIP (300 mil "skinny DIP")
  - 28-pin PLCC
- 87C751 available in erasable quartz lid or one-time programmable plastic packages
- Wide oscillator frequency range
- Low power consumption:
  - Normal operation: less than 11mA @ 5V, 12MHz
  - Idle mode
  - Power-down mode
- $2k \times 8$  ROM (83C751)
- $2k \times 8$  EPROM (87C751)
- $64 \times 8$  RAM
- 16-bit auto reloadable counter/timer
- Fixed-rate timer
- Boolean processor
- CMOS and TTL compatible
- Well suited for logic replacement, consumer and industrial applications
- LED drive outputs

## PIN CONFIGURATIONS



## ORDERING INFORMATION

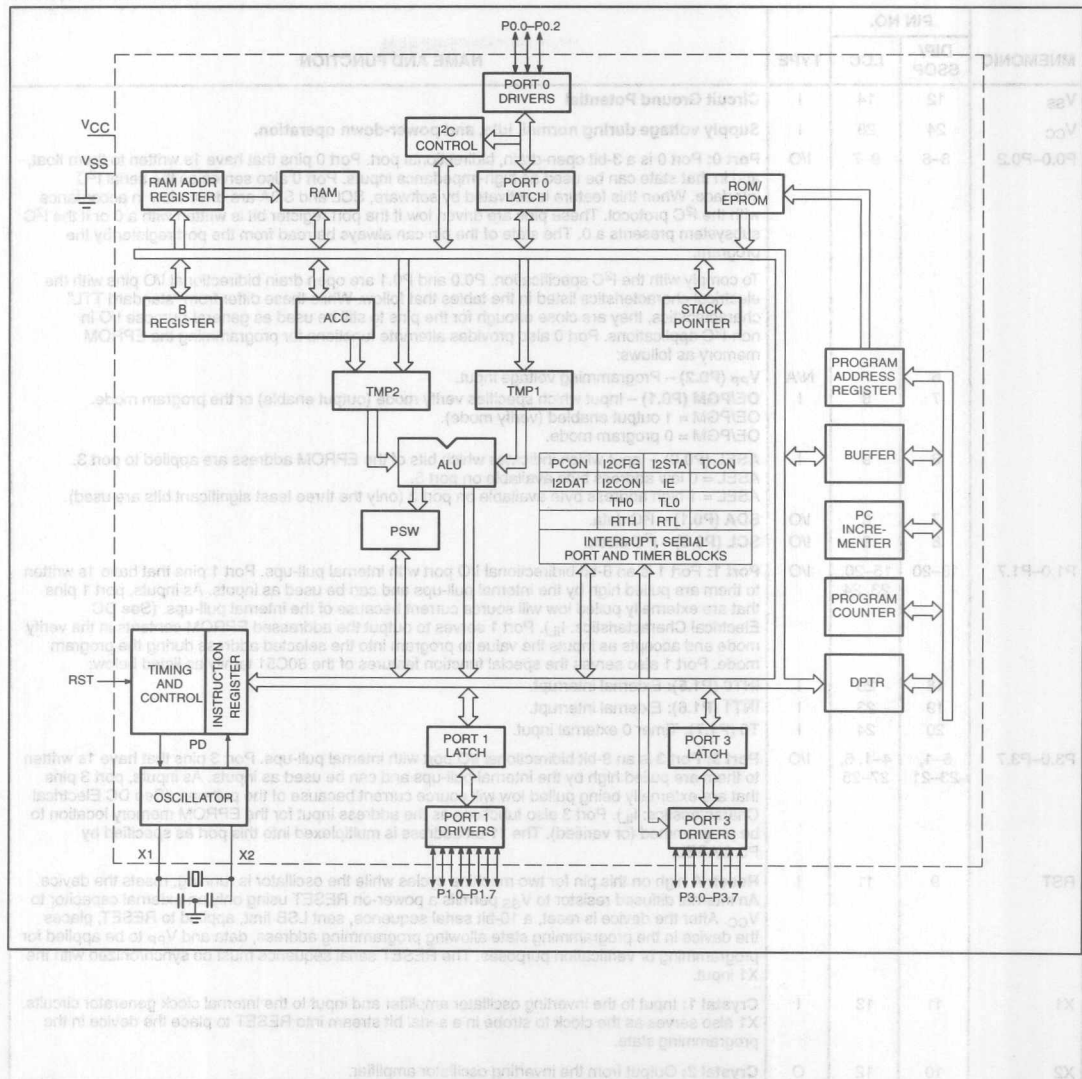
ROM	EPROM <sup>1</sup>		TEMPERATURE RANGE °C AND PACKAGE	FREQUENCY MHz	DRAWING NUMBER
	S87C751-1F24	UV	0 to +70, Ceramic Dual In-line Package	3.5 to 12	0586B
	S87C751-2F24	UV	-40 to +85, Ceramic Dual In-line Package	3.5 to 12	0586B
	S87C751-4F24	UV	0 to +70, Ceramic Dual In-line Package	3.5 to 16	0586B
	S87C751-5F24	UV	-40 to +85, Ceramic Dual In-line Package	3.5 to 16	0586B
S83C751-1N24	S87C751-1N24	OTP	0 to +70, Plastic Dual In-line Package	3.5 to 12	0410D
S83C751-2N24	S87C751-2N24	OTP	-40 to +85, Plastic Dual In-line Package	3.5 to 12	0410D
S83C751-4N24	S87C751-4N24	OTP	0 to +70, Plastic Dual In-line Package	3.5 to 16	0410D
S83C751-5N24	S87C751-5N24	OTP	-40 to +85, Plastic Dual In-line Package	3.5 to 16	0410D
S83C751-1A28	S87C751-1A28	OTP	0 to +70, Plastic Leaded Chip Carrier	3.5 to 12	0401F
S83C751-2A28	S87C751-2A28	OTP	-40 to +85, Plastic Leaded Chip Carrier	3.5 to 12	0401F
S83C751-4A28	S87C751-4A28	OTP	0 to +70, Plastic Leaded Chip Carrier	3.5 to 16	0401F
S83C751-5A28	S87C751-5A28	OTP	-40 to +85, Plastic Leaded Chip Carrier	3.5 to 16	0401F

### NOTE:

1. OTP = One Time Programmable EPROM. UV = UV Erasable EPROM.

## CMOS single-chip 8-bit microcontroller 83C751/87C751

## BLOCK DIAGRAM



## CMOS single-chip 8-bit microcontroller 80C751/87C751

## PIN DESCRIPTIONS

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP/SSOP	LCC		
V <sub>SS</sub>	12	14	I	<b>Circuit Ground Potential</b>
V <sub>CC</sub>	24	28	I	<b>Supply voltage during normal, idle, and power-down operation.</b>
P0.0–P0.2	8–6	9–7	I/O	<p><b>Port 0:</b> Port 0 is a 3-bit open-drain, bidirectional port. Port 0 pins that have 1s written to them float, and in that state can be used as high-impedance inputs. Port 0 also serves as the serial I<sup>2</sup>C interface. When this feature is activated by software, SCL and SDA are driven low in accordance with the I<sup>2</sup>C protocol. These pins are driven low if the port register bit is written with a 0 or if the I<sup>2</sup>C subsystem presents a 0. The state of the pin can always be read from the port register by the program.</p> <p>To comply with the I<sup>2</sup>C specification, P0.0 and P0.1 are open drain bidirectional I/O pins with the electrical characteristics listed in the tables that follow. While these differ from "standard TTL" characteristics, they are close enough for the pins to still be used as general-purpose I/O in non-I<sup>2</sup>C applications. Port 0 also provides alternate functions for programming the EPROM memory as follows:</p> <p><b>V<sub>PP</sub> (P0.2)</b> – Programming voltage input.</p> <p><b>OE/PGM (P0.1)</b> – Input which specifies verify mode (output enable) or the program mode. OE/PGM = 1 output enabled (verify mode). OE/PGM = 0 program mode.</p> <p><b>ASEL (P0.0)</b> – Input which indicates which bits of the EPROM address are applied to port 3. ASEL = 0 low address byte available on port 3. ASEL = 1 high address byte available on port 3 (only the three least significant bits are used).</p> <p><b>SDA (P0.1)</b> – I<sup>2</sup>C data.</p> <p><b>SCL (P0.0)</b> – I<sup>2</sup>C clock.</p>
P1.0–P1.7	6	7	N/A	
	7	8	I	
	8	9	I/O	
	7	8	I/O	
	8	9	I/O	
P3.0–P3.7	13–20	15–20, 23, 24	I/O	<p><b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I<sub>IL</sub>). Port 1 serves to output the addressed EPROM contents in the verify mode and accepts as inputs the value to program into the selected address during the program mode. Port 1 also serves the special function features of the 80C51 family as listed below:</p> <p><b>INT0 (P1.5):</b> External interrupt.</p> <p><b>INT1 (P1.6):</b> External interrupt.</p> <p><b>T0 (P1.7):</b> Timer 0 external input.</p>
	18	20	I	
	19	23	I	
	20	24	I	
	5–1, 23–21	4–1, 6, 27–25	I/O	<p><b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: I<sub>IL</sub>). Port 3 also functions as the address input for the EPROM memory location to be programmed (or verified). The 11-bit address is multiplexed into this port as specified by P0.0/ASEL.</p>
RST	9	11	I	<p><b>Reset:</b> A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to V<sub>SS</sub> permits a power-on RESET using only an external capacitor to V<sub>CC</sub>. After the device is reset, a 10-bit serial sequence, sent LSB first, applied to RESET, places the device in the programming state allowing programming address, data and V<sub>PP</sub> to be applied for programming or verification purposes. The RESET serial sequence must be synchronized with the X1 input.</p>
X1	11	13	I	<p><b>Crystal 1:</b> Input to the inverting oscillator amplifier and input to the internal clock generator circuits. X1 also serves as the clock to strobe in a serial bit stream into RESET to place the device in the programming state.</p>
X2	10	12	O	<p><b>Crystal 2:</b> Output from the inverting oscillator amplifier.</p>

## CMOS single-chip 8-bit microcontroller 83C751/87C751

ABSOLUTE MAXIMUM RATINGS<sup>1, 2</sup>

PARAMETER	RATING	UNIT
Storage temperature range	-65 to +150	°C
Voltage from $V_{CC}$ to $V_{SS}$	-0.5 to +6.5	V
Voltage from any pin to $V_{SS}$ (except $V_{PP}$ )	-0.5 to $V_{CC} + 0.5$	V
Power dissipation	1.0	W
Voltage on $V_{PP}$ pin to $V_{SS}$	0 to +13.0	V
Maximum $I_{OL}$ per I/O pin	10	mA

## NOTES:

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
- This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.

## DC ELECTRICAL CHARACTERISTICS

$T_{amb} = 0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$  or  $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$  for 87C751,  $V_{CC} = 5\text{V} \pm 20\%$  for 83C751,  $V_{SS} = 0\text{V}$ <sup>1</sup>

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			MIN	MAX	
$V_{IL}$	Input low voltage, except SDA, SCL		-0.5	$0.2V_{DD} - 0.1$	V
$V_{IH}$	Input high voltage, except X1, RST		$0.2V_{CC} + 0.9$	$V_{CC} + 0.5$	V
$V_{IH1}$	Input high voltage, X1, RST		$0.7V_{CC}$	$V_{CC} + 0.5$	V
$V_{IL1}$	SDA, SCL, P0.2 Input low voltage		-0.5	$0.3V_{CC}$	V
$V_{IH2}$	Input high voltage		$0.7V_{CC}$	$V_{CC} + 0.5$	V
$V_{OL}$	Output low voltage, ports 1 and 3	$I_{OL} = 1.6\text{mA}^2$		0.45	V
$V_{OL1}$	Output low voltage, port 0.2	$I_{OL} = 3.2\text{mA}^2$		0.45	V
$V_{OH}$	Output high voltage, ports 1 and 3	$I_{OH} = -60\mu\text{A}$ $I_{OH} = -25\mu\text{A}$ $I_{OH} = -10\mu\text{A}$	2.4 $0.75V_{CC}$ $0.9V_{CC}$		V
$V_{OL2}$	Port 0.0 and 0.1 ( $I^2C$ ) – Drivers Output low voltage	$I_{OL} = 3\text{mA}$ (over $V_{CC}$ range)		0.4	V
C	Capacitance			10	pF
$I_{IL}$	Logical 0 input current, ports 1 and 3	$V_{IN} = 0.45\text{V}$		-50	$\mu\text{A}$
$I_{TL}$	Logical 1 to 0 transition current, ports 1 and 3 <sup>3</sup>	$V_{IN} = 2\text{V}$ (0 to $70^{\circ}\text{C}$ ) $V_{IN} = 2\text{V}$ ( $-40$ to $+85^{\circ}\text{C}$ )		-650 -750	$\mu\text{A}$
$I_{LI}$	Input leakage current, port 0	$0.45 < V_{IN} < V_{CC}$		$\pm 10$	$\mu\text{A}$
R <sub>RST</sub>	Internal pull-down resistor		25	175	k $\Omega$
$C_{IO}$	Pin capacitance	Test freq = 1MHz, $T_{amb} = 25^{\circ}\text{C}$		10	pF
$I_{PD}$	Power-down current <sup>4</sup>	$V_{CC} = 2$ to $V_{CC}$ max		50	$\mu\text{A}$
$V_{PP}$	$V_{PP}$ program voltage (for 87C751 only)	$V_{SS} = 0\text{V}$ $V_{CC} = 5\text{V} \pm 10\%$ $T_{amb} = 21^{\circ}\text{C}$ to $27^{\circ}\text{C}$	12.5	13.0	V
$I_{PP}$	Program current (for 87C751 only)	$V_{PP} = 13.0\text{V}$		50	mA
$I_{CC}$	Supply current (see Figure 2)				

NOTES TO DC ELECTRICAL CHARACTERISTICS ON NEXT PAGE



## CMOS single-chip 8-bit microcontroller 83C751/87C751

**NOTES TO DC ELECTRICAL CHARACTERISTICS:**

- NOTES TO DC ELECTRICAL CHARACTERISTICS:**
- Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to  $V_{SS}$  unless otherwise noted.
  - Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:
 

Maximum $I_{OL}$ per port pin:	10mA	(NOTE: This is 85°C spec.)
Maximum $I_{OL}$ per 8-bit port:	26mA	
Maximum total $I_{OL}$ for all outputs:	67mA	

 If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
  - Pins of ports 1 and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when  $V_{IH}$  is approximately 2V.
  - Power-down  $I_{CC}$  is measured with all output pins disconnected; port 0 =  $V_{CC}$ ; X2, X1 n.c.; RST =  $V_{SS}$ .
  - Active  $I_{CC}$  is measured with all output pins disconnected; X1 driven with  $t_{CLCH}$ ,  $t_{CHCL}$  = 5ns,  $V_{IL}$  =  $V_{SS} + 0.5V$ ,  $V_{IH}$  =  $V_{CC} - 0.5V$ ; X2 n.c.; RST = port 0 =  $V_{CC}$ .  $I_{CC}$  will be slightly higher if a crystal oscillator is used.
  - Idle  $I_{CC}$  is measured with all output pins disconnected; X1 driven with  $t_{CLCH}$ ,  $t_{CHCL}$  = 5ns,  $V_{IL}$  =  $V_{SS} + 0.5V$ ,  $V_{IH}$  =  $V_{CC} - 0.5V$ ; X2 n.c.; port 0 =  $V_{CC}$ ; RST =  $V_{SS}$ .

## AC ELECTRICAL CHARACTERISTICS

$T_{amb} = 0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$  or  $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$  for 87C751,  $V_{CC} = 5\text{V} \pm 20\%$  for 83C751,  $V_{SS} = 0\text{V}^{1,2}$

SYMBOL	PARAMETER	12MHz CLOCK		VARIABLE CLOCK		UNIT
		MIN	MAX	MIN	MAX	
1/t <sub>CLCL</sub>	Oscillator frequency:			3.5	12	MHz
				3.5	16	MHz
<b>External Clock (Figure 1)</b>						
t <sub>CHCX</sub>	High time	20		20		ns
t <sub>CLCX</sub>	Low time	20		20		ns
t <sub>CLCH</sub>	Rise time		20		20	ns
t <sub>CHCL</sub>	Fall time		20		20	ns

NOTES:

1. Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to  $V_{SS}$  unless otherwise noted.
2. Load capacitance for ports = 80pF.

## CMOS single-chip 8-bit microcontroller 83C751/87C751

## EXPLANATION OF THE AC SYMBOLS

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

C – Clock

D – Input data

H – Logic level high

L – Logic level low

Q – Output data

T – Time

V – Valid

X – No longer a valid logic level

Z – Float

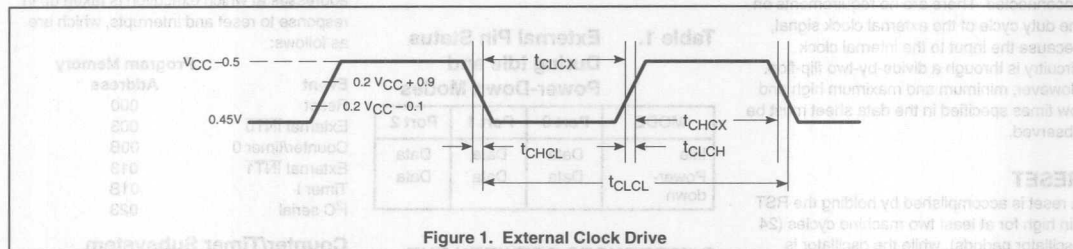
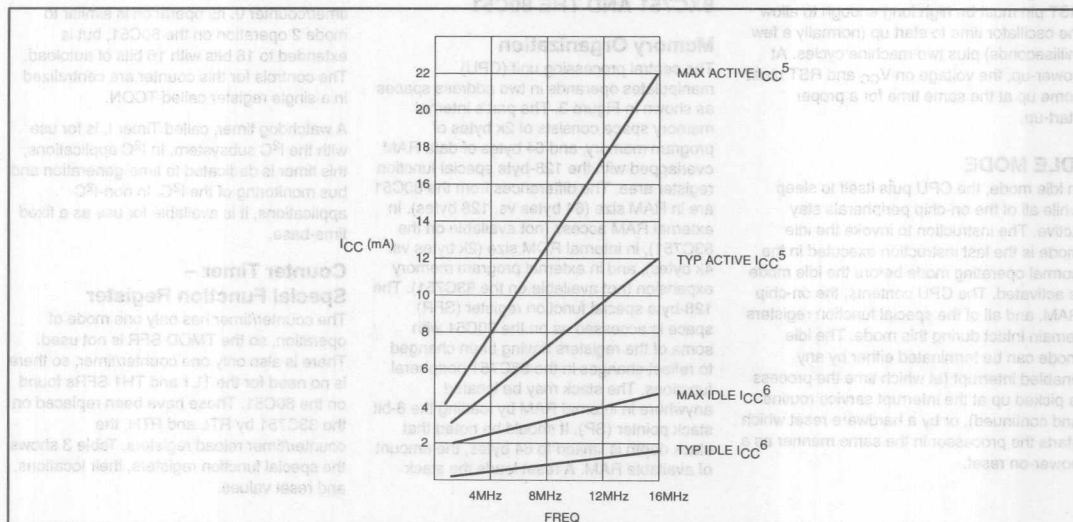


Figure 1. External Clock Drive

Figure 2.  $I_{CC}$  vs. FREQ

Maximum  $I_{CC}$  values taken at  $V_{CC}$  max and worst case temperature.

Typical  $I_{CC}$  values taken at  $V_{CC} = 5.0V$  and  $25^{\circ}C$ .

Notes 5 and 6 refer to DC Electrical Characteristics.

## CMOS single-chip 8-bit microcontroller 83C751/87C751

**OSCILLATOR CHARACTERISTICS**

X1 and X2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator.

To drive the device from an external clock source, X1 should be driven while X2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

**RESET**

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. To insure a good power-up reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few milliseconds) plus two machine cycles. At power-up, the voltage on V<sub>CC</sub> and RST must come up at the same time for a proper start-up.

**IDLE MODE**

In idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

**POWER-DOWN MODE**

In the power-down mode, the oscillator is stopped and the instruction to invoke power-down is the last instruction executed. Only the contents of the on-chip RAM are preserved. A hardware reset is the only way to terminate the power-down mode: the control bits for the reduced power modes are in the special function register PCON.

**Table 1. External Pin Status During Idle and Power-Down Modes**

MODE	Port 0	Port 1	Port 2
Idle	Data	Data	Data
Power-down	Data	Data	Data

**DIFFERENCES BETWEEN THE 83C751 AND THE 80C51****Memory Organization**

The central processing unit (CPU) manipulates operands in two address spaces as shown in Figure 3. The part's internal memory space consists of 2k bytes of program memory, and 64 bytes of data RAM overlapped with the 128-byte special function register area. The differences from the 80C51 are in RAM size (64 bytes vs. 128 bytes), in external RAM access (not available on the 83C751), in internal ROM size (2k bytes vs. 4k bytes), and in external program memory expansion (not available on the 83C751). The 128-byte special function register (SFR) space is accessed as on the 80C51 with some of the registers having been changed to reflect changes in the 83C751 peripheral functions. The stack may be located anywhere in internal RAM by loading the 8-bit stack pointer (SP). It should be noted that stack depth is limited to 64 bytes, the amount of available RAM. A reset loads the stack

pointer with 07 (which is pre-incremented on a PUSH instruction).

**Program Memory**

On the 83C751, program memory is 2048 bytes long and is not externally expandable, so the 80C51 instructions MOVX, LJMP, and LCALL are not implemented. The only fixed locations in program memory are the addresses at which execution is taken up in response to reset and interrupts, which are as follows:

Event	Program Memory Address
Reset	000
External INT0	003
Counter/timer 0	00B
External INT1	013
Timer 1	01B
I <sup>2</sup> C serial	023

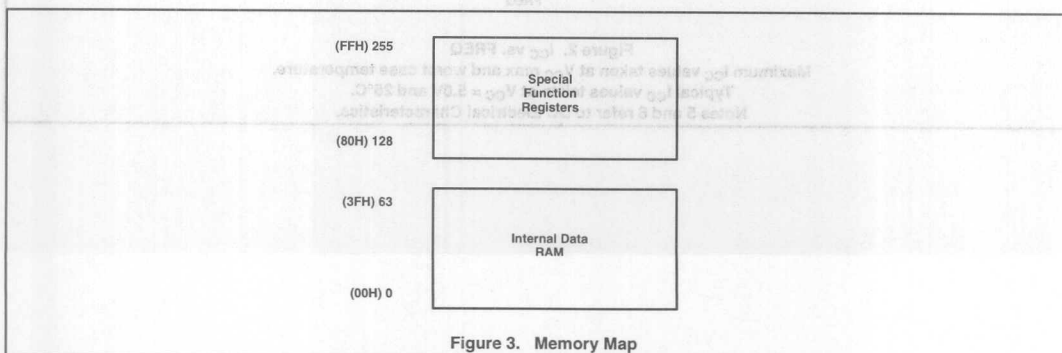
**Counter/Timer Subsystem**

The 83C751 has one counter/timer called timer/counter 0. Its operation is similar to mode 2 operation on the 80C51, but is extended to 16 bits with 16 bits of autoloader. The controls for this counter are centralized in a single register called TCON.

A watchdog timer, called Timer 1, is for use with the I<sup>2</sup>C subsystem. In I<sup>2</sup>C applications, this timer is dedicated to time-generation and bus monitoring of the I<sup>2</sup>C. In non-I<sup>2</sup>C applications, it is available for use as a fixed time-base.

**Counter Timer – Special Function Register**

The counter/timer has only one mode of operation, so the TMOD SFR is not used. There is also only one counter/timer, so there is no need for the TL1 and TH1 SFRs found on the 80C51. These have been replaced on the 83C751 by RTL and RTH, the counter/timer reload registers. Table 3 shows the special function registers, their locations, and reset values.



## CMOS single-chip 8-bit microcontroller 83C751/87C751

**Interrupt Subsystem – Fixed****Priority**

The IP register and the 2-level interrupt system of the 80C51 are eliminated. Simultaneous interrupt conditions are resolved by a single-level, fixed priority as follows:

Highest priority: Pin INT0  
Counter/timer flag 0  
Pin INT1  
Timer 1  
Serial I<sup>2</sup>C

Lowest priority: Serial I<sup>2</sup>C

**Special Function Register – Interrupt Subsystem**

Because the interrupt structure is single level on the 83C751, there is no need for the IP SFR, so it is not used.

**Serial Communications**

The 83C751 contains an I<sup>2</sup>C serial communications port instead of the 80C51 UART. The I<sup>2</sup>C serial port is a single bit

hardware interface with all of the hardware necessary to support multimaster and slave operations. Also included are receiver digital filters and timer (timer 1) for communication watch-dog purposes. The I<sup>2</sup>C serial port is controlled through four special function registers; I<sup>2</sup>C control, I<sup>2</sup>C data, I<sup>2</sup>C status, and I<sup>2</sup>C configuration.

**Special Function Register – Serial Communications**

The 83C751 contains many of the special function registers (SFR) that are found on the 80C51. Due to the different peripheral features on the 83C751, there are several additional SFRs and several that have been changed.

Since the standard UART found on the 80C51 has been replaced by the I<sup>2</sup>C serial interface, the UART SFRs, SCON, and SBUF have been replaced by I2CON and I2DAT, and two additional I<sup>2</sup>C registers have been added (I2STA and I2CFG).

**I/O Port Latches (P0, P1, P3)**

The port latches function the same as those on the 80C51. Since there is no port 2 on the 83C751, the P2 latch is not used. Port 0 on the 83C751 has only 3 bits, so only 3 bits of the P0 SFR have a useful function.

**Special Function Register – I/O Port Latches**

There is no Port2 on the 83C751, so P2 is not used. Also, only 3 bits of P0 SFR have a useful function.

**Data Pointer (DPTR)**

The data pointer (DPTR) consists of a high byte (DPH) and a low byte (DPL). In the 80C51 this register allows the access of external data memory using the MOVX instruction. Since the 83C751 does not support MOVX or external memory accesses, this register is generally used as a 16-bit offset pointer of the accumulator in a MOVX instruction. DPTR may also be manipulated as two independent 8-bit registers.

**Table 2. I<sup>2</sup>C Special Function Register Addresses**

REGISTER ADDRESS				BIT ADDRESS							
NAME	SYMBOL	ADDRESS	MSB	LSB							
I <sup>2</sup> C control	I2CON	98	9F	9E	9D	9C	9B	9A	99	98	
I <sup>2</sup> C data	I2DAT	99	—	—	—	—	—	—	—	—	
I <sup>2</sup> C configuration	I2CFG	D8	DF	DE	DD	DC	DB	DA	D9	D8	
I <sup>2</sup> C status	I2STA	F8	FF	FE	FD	FC	FB	FA	F9	F8	

**Table 3. 8XC751 Special Function Registers**

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB								
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
DPTR:	Data pointer (2 bytes)										
DPH	High byte	83H									00H
DPL	Low byte	82H									00H
			DF	DE	DD	DC	DB	DA	D9	D8	
I <sup>2</sup> CFG*#	I <sup>2</sup> C configuration	D8H/RD	SLAVEN	MASTRO	0	TIRUN	—	—	CT1	CT0	0000xx00B
		WR	SLAVEN	MASTRO	CLRTI	TIRUN	—	—	CT1	CT0	
			9F	9E	9D	9C	9B	9A	99	98	
I <sup>2</sup> CON*#	I <sup>2</sup> C control	98H/RD	RDAT	ATN	DRDY	ARL	STR	STP	MASTER	—	81H
		WR	CXA	IDLE	CDR	CARL	CSTR	CSTP	XSTR	XSTP	
			RDAT	0	0	0	0	0	0	0	80H
I <sup>2</sup> DAT#	I <sup>2</sup> C data	99H/RD	XDAT	X	X	X	X	X	X	X	
		WR									
			FF	FE	FD	FC	FB	FA	F9	F8	
I <sup>2</sup> STA*#	I <sup>2</sup> C control	F8H	—	IDLE	XDATA	XACTV	MAKSTR	MAKSTP	XSTR	XSTP	x0100000B
			AF	AE	AD	AC	AB	AA	A9	A8	
IE*#	Interrupt enable	A8H	EA	—	—	EI2	ETI	EX1	ET0	EX0	00H
								82	81	80	
P0*#	Port 0	80H	—	—	—	—	—	—	SDA	SCL	xxxxx111B
			97	96	95	94	93	92	91	90	
P1*	Port 1	90H	T0	INT1	INT0	—	—	—	—	—	FFH
P3*	Port 3	B0H	B7	B6	B5	B4	B3	B2	B1	B0	FFH
PCON#	Power control	87H	—	—	—	—	—	—	PD	IDL	xxxxxx00B
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program status word	D0H	CY	AC	F0	RS1	RS0	OV	—	P	00H
SP	Stack pointer	81H									07H
			8F	8E	8D	8C	8B	8A	89	88	
TCON*#	Timer/counter control	88H	GATE	C/T	TF	TR	IE0	IT0	IE1	IT1	00H
TL#	Timer low byte	8AH									00H
TH#	Timer high byte	8CH									00H
RTL#	Timer low reload	8BH									00H
RTH#	Timer high reload	8DH									00H

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.



1  
ree  
Each  
ister  
input

RTL. The value in the reload registers is left unchanged. See the 83C751 counter/timer block diagram in Figure 4. The TF bit in special function register TCON is set on counter overflow and, if the interrupt is enabled, will generate an interrupt.

## TCON Register

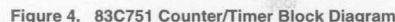
GATE	C/T	TF	TR	IE0	IT0	IE1	IT1
MSB				LSB			

These flags are functionally identical to the corresponding 80C51 flags, except that there is only one timer on the 83C751 and the flags are therefore combined into one register.

The I<sup>2</sup>C watchdog timer, timer 1, is also available as a general-purpose fixed-rate timer when the I<sup>2</sup>C interface is not being used. A clock rate of 1/12 the oscillator frequency forms the input to the timer. Timer 1 has a timeout interval of 1024 machine cycles when used as a fixed-rate timer.

The I<sup>2</sup>C bus uses two wires (SDA and SCL) to transfer information between devices connected to the bus. The main features of the bus are:

- A large family of I<sup>2</sup>C compatible ICs is available. See the I<sup>2</sup>C section of this manual for more details on the bus and available ICs.



## CMOS single-chip 8-bit microcontroller 83C751/87C751

The 83C751 I<sup>2</sup>C subsystem includes hardware to simplify the software required to drive the I<sup>2</sup>C bus. The hardware is a single-bit interface which in addition to including the necessary arbitration and framing error checks, includes clock stretching and a bus timeout timer. The interface is synchronized to software either through polled loops or interrupts. Refer to the application note AN422, in Section 4, entitled "Using the 8XC751 Microcontroller as an I<sup>2</sup>C Bus Master" for additional discussion of the 83C751 I<sup>2</sup>C interface and sample driver routines.

Six time spans are important in I<sup>2</sup>C operation and are insured by timer I:

- The MINIMUM HIGH time for SCL when this device is the master.
- The MINIMUM LOW time for SCL when this device is a master. This is not very important for a single-bit hardware interface like this one, because the SCL low time is stretched until the software responds to the I<sup>2</sup>C flags. The software response time normally meets or exceeds the MIN LO time. In cases where the software responds within MIN HI + MIN LO time, timer I will ensure that the minimum time is met.
- The MINIMUM SCL HIGH TO SDA HIGH time in a stop condition.
- The MINIMUM SDA HIGH TO SDA LOW time between I<sup>2</sup>C stop and start conditions (4.7µs, see spec.).
- The MINIMUM SDA LOW TO SCL LOW time in a start condition.
- The MAXIMUM SCL CHANGE time while an I<sup>2</sup>C frame is in progress. A frame is in progress between a start condition and the following stop condition. This time span serves to detect a lack of software response on this 8XC751 as well as external I<sup>2</sup>C problems. SCL "stuck low" indicates a faulty master or slave. SCL "stuck high" may mean a faulty device, or that noise induced onto the I<sup>2</sup>C bus caused all masters to withdraw from I<sup>2</sup>C arbitration.

The first five of these times are 4.7µs (see I<sup>2</sup>C specification) and are covered by the low order three bits of timer I. Timer I is clocked by the 8XC751 oscillator, which can vary in frequency from 0.5 to 16MHz. Timer I can be preloaded with one of four values to optimize timing for different oscillator frequencies. At lower frequencies, software response time is increased and will degrade maximum performance of the I<sup>2</sup>C bus. See special function register I2CFG description for prescale values (CT0, CT1).

The MAXIMUM SCL CHANGE time is important, but its exact span is not critical. The complete 10 bits of timer I are used to

count out the maximum time. When I<sup>2</sup>C operation is enabled, this counter is cleared by transitions on the SCL pin. The timer does not run between I<sup>2</sup>C frames (i.e., whenever reset or stop occurred more recently than the last start). When this counter is running, it will carry out after 1020 to 1023 machine cycles have elapsed since a change on SCL. A carry out causes a hardware reset of the 83C751 I<sup>2</sup>C interface and generates an interrupt if the timer I interrupt is enabled. In cases where the bus hangup is due to a lack of software response by this 83C751, the reset releases SCL and allows I<sup>2</sup>C operation among other devices to continue.

#### I<sup>2</sup>C Interrupts

If I<sup>2</sup>C interrupts are enabled (EA and EI2 are both set to 1), an I<sup>2</sup>C interrupt will occur whenever the ATN flag is set by a start, stop, arbitration loss, or data ready condition (refer to the description of ATN following). In practice, it is not efficient to operate the I<sup>2</sup>C interface in this fashion because the I<sup>2</sup>C interrupt service routine would somehow have to distinguish between hundreds of possible conditions. Also, since I<sup>2</sup>C can operate at a fairly high rate, the software may execute faster if the code simply waits for the I<sup>2</sup>C interface.

Typically, the I<sup>2</sup>C interrupt should only be used to indicate a start condition at an idle slave device, or a stop condition at an idle master device (if it is waiting to use the I<sup>2</sup>C bus). This is accomplished by enabling the I<sup>2</sup>C interrupt only during the aforementioned conditions.

#### I<sup>2</sup>C Register I2CON

	7	6	5	4	3	2	1	0
Read	RDAT	ATN	DRDY	ARL	STR	STP	MASTER	—
Write	CXA	IDLE	CDR	CARL	CSTR	CSTP	XSTR	XSTP

#### Reading I2CON

**RDAT** The data from SDA is captured into "Receive Data" whenever a rising edge occurs on SCL. RDAT is also available (with seven low-order zeros) in the I2DAT register. The difference between reading it here and there is that reading I2DAT clears DRDY, allowing the I<sup>2</sup>C to proceed on to another bit. Typically, the first seven bits of a received byte are read from I2DAT, while the 8th is read here. Then I2DAT can be written to send the Ack bit and clear DRDY.

**ATN** "ATTeNtion" is 1 when one or more of DRDY, ARL, STR, or STP is 1. Thus, ATN comprises a single bit that can be tested to release the I<sup>2</sup>C service routine from a "wait loop."

**DRDY** "Data Ready" (and thus ATN) is set when a rising edge occurs on SCL, except at idle slave. DRDY is cleared by writing CDR = 1, or by writing or reading the I2DAT register. The following low period on SCL is stretched until the program responds by clearing DRDY.

#### Checking ATN and DRDY

When a program detects ATN = 1, it should next check DRDY. If DRDY = 1, then if it receives the last bit, it should capture the data from RDAT (in I2DAT or I2CON). Next, if the next bit is to be sent, it should be written to I2DAT. One way or another, it should clear DRDY and then return to monitoring ATN. Note that if any of ARL, STR, or STP is set, clearing DRDY will not release SCL to high, so that the I<sup>2</sup>C will not go on to the next bit. If a program detects ATN = 1, and DRDY = 0, it should go on to examine ARL, STR, and STP.

**ARL** "Arbitration Loss" is 1 when transmit Active was set, but this 83C751 lost arbitration to another transmitter. Transmit Active is cleared when ARL is 1. There are four separate cases in which ARL is set.

1. If the program sent a 1 or repeated start, but another device sent a 0, or a stop, so that SDA is 0 at the rising edge of SCL. (If the other device sent a stop, the setting of ARL will be followed shortly by STP being set.)
2. If the program sent a 1, but another device sent a repeated start, and it drove SDA low before the 83C751 could drive SCL low. (This type of ARL is always accompanied by STR = 1.)
3. In master mode, if the program sent a repeated start, but another device sent a 1, and it drove SCL low before this 83C751 could drive SDA low.
4. In master mode, if the program sent stop, but it could not be sent because another device sent a 0.

**STR** "STaRt" is set to a 1 when an I<sup>2</sup>C start condition is detected at a non-idle slave or at a master. (STR is not set when an idle slave becomes active due to a start bit; the slave has nothing useful to do until the rising edge of SCL sets DRDY.)

## CMOS single-chip 8-bit microcontroller 83C751/87C751

**STP** "SToP" is set to 1 when an I<sup>2</sup>C stop condition is detected at a non-idle slave or at a master. (STP is not set for a stop condition at an idle slave.)

**MASTER** "MASTER" is 1 if this 83C751 is currently a master on the I<sup>2</sup>C. MASTER is set when MASTRQ is 1 and the bus is not busy (i.e., if a start bit hasn't been received since reset or a "Timer I" time-out, or if a stop has been received since the last start). MASTER is cleared when ARL is set, or after the software writes MASTRQ = 0 and then XSTP = 1.

#### Writing I2CON

Typically, for each bit in an I<sup>2</sup>C message, a service routine waits for ATN = 1. Based on DRDY, ARL, STR, and STP, and on the current bit position in the message, it may then write I2CON with one or more of the following bits, or it may read or write the I2DAT register.

**CXA** Writing a 1 to "Clear Xmit Active" clears the Transmit Active state. (Reading the I2DAT register also does this.)

#### Regarding Transmit Active

Transmit Active is set by writing the I2DAT register, or by writing I2CON with XSTR = 1 or XSTP = 1. The I<sup>2</sup>C interface will only drive the SDA line low when Transmit Active is set, and the ARL bit will only be set to 1 when Transmit Active is set. Transmit Active is cleared by reading the I2DAT register, or by writing I2CON with CXA = 1. Transmit Active is automatically cleared when ARL is 1.

**IDLE** Writing 1 to "IDLE" causes a slave's I<sup>2</sup>C hardware to ignore the I<sup>2</sup>C until the next start condition (but if MASTRQ is 1, then a stop condition will make the 83C751 into a master).

**CDR** Writing a 1 to "Clear Data Ready" clears DRDY. (Reading or writing the I2DAT register also does this.)

**CARL** Writing a 1 to "Clear Arbitration Loss" clears the ARL bit.

**CSTR** Writing a 1 to "Clear STaRt" clears the STR bit.

**CSTP** Writing a 1 to "Clear SToP" clears the STP bit. Note that if one or more of DRDY, ARL, STR, or STP is 1, the low time of SCL is stretched until the service routine responds by clearing them.

**XSTR** Writing 1s to "Xmit repeated STaRt" and CDR tells the I<sup>2</sup>C hardware to send a repeated start condition.

This should only be at a master. Note that XSTR need not and should not be used to send an "initial" (nonrepeated) start; it is sent automatically by the I<sup>2</sup>C hardware. Writing XSTR = 1 includes the effect of writing I2DAT with XDAT = 1; it sets Transmit Active and releases SDA to high during the SCL low time. After SCL goes high, the I<sup>2</sup>C hardware waits for the suitable minimum time and then drives SDA low to make the start condition.

**XSTP** Writing 1s to "Xmit SToP" and CDR tells the I<sup>2</sup>C hardware to send a stop condition. This should only be done at a master. If there are no more messages to initiate, the service routine should clear the MASTRQ bit in I2CFG to 0 before writing XSTP with 1. Writing XSTP = 1 includes the effect of writing I2DAT with XDAT = 0; it sets Transmit Active and drives SDA low during the SCL low time. After SCL goes high, the I<sup>2</sup>C hardware waits for the suitable minimum time and then releases SDA to high to make the stop condition.

**NOTE:** Because of the manner in which register bit addressing is implemented in the 80C51 family, the I2CON register should never be altered by use of the SETB, CLR, CPL, MOV (bit), or JBC instructions. This is due to the fact that read and write functions of this register are different. Testing of I2CON bits via the JB and JNB instructions is supported.

#### I<sup>2</sup>C Register I2DAT

	7	6	5	4	3	2	1	0
Read	RDAT	0	0	0	0	0	0	0
Write	XDAT	X	X	X	X	X	X	X

**RDAT** "Receive DATA" is captured from SDA every rising edge of SCL. Reading I2DAT also clears DRDY and the Transmit Active state.

**XDAT** "Xmit Data" sets the data for the next bit. Writing I2DAT also clears DRDY and sets the Transmit Active state.

#### Regarding Software Response Time

Because the 83C751 can run at 16MHz, and because the I<sup>2</sup>C interface is optimized for high-speed operation, it is quite likely that an I<sup>2</sup>C service routine will sometimes respond to DRDY (which is set at a rising edge of SCL) and write I2DAT before SCL has gone low again. If XDAT were applied directly to SDA, this situation would produce an I<sup>2</sup>C protocol violation. The programmer need not worry

about this possibility because XDAT is applied to SDA only when SCL is low.

Conversely, a program that includes an I<sup>2</sup>C service routine may take a long time to respond to DRDY. Typically, an I<sup>2</sup>C routine operates on a flag-polling basis during a message, with interrupts from other peripheral functions enabled. If an interrupt occurs, it will delay the response of the I<sup>2</sup>C service routine. The programmer need not worry about this very much either, because the I<sup>2</sup>C hardware stretches the SCL low time until the service routine responds. The only constraint on the response is that it must not exceed the Timer I time-out, which is at least 765 microseconds.

#### I<sup>2</sup>C Register I2CFG

	7	6	5	4	3	2	1	0
Read	SLAVEN	MASTRQ	0	TIRUN	-	-	CT1	CT0
Write	SLAVEN	MASTRQ	CLRTI	TIRUN	-	-	CT1	CT0

**SLAVEN** Writing a 1 to "SLAVE Enable" enables the slave functions of the I<sup>2</sup>C subsystem. If SLAVEN and MASTRQ are 0, the I<sup>2</sup>C hardware is disabled. This bit is cleared to 0 by reset and by an I<sup>2</sup>C time-out.

**MASTRQ** Writing a 1 to "MASTRQ" requests mastership of the I<sup>2</sup>C. If a frame from another master is in progress when this bit is changed from 0 to 1, action is delayed until a stop condition is detected. Then, or immediately if a frame is not in progress, a start condition is sent and DRDY is set (thus making ATN 1 and generating an I<sup>2</sup>C interrupt). When a master wishes to release mastership status of the I<sup>2</sup>C, it writes a 1 to XSTP in I2CON. MASTRQ is cleared by reset and by an I<sup>2</sup>C time-out.

**CLRTI** Writing a 1 to this bit clears the Timer I interrupt flag. This bit position always reads as a 0.

**TIRUN** Writing a 1 to this bit lets Timer I run; a zero stops and clears it. Together with SLAVEN, MASTRQ, and MASTER, this bit determines operational modes as shown in Table 4.

**CT1,0** These two bits are programmed as a function of the OSC rate, to optimize the MIN HI and LO time of SCL when this 83C751 is a master on the I<sup>2</sup>C. The time value determined by these bits controls both of these parameters, and also the timing for stop and start conditions. These bits are cleared to 00 by reset.

**Table 4. Interaction of TIRUN with SLAVEN, MASTRQ, and MASTER**

SLAVEN, MASTRQ, MASTER	TIRUN	OPERATING MODE
All 0	0	The I <sup>2</sup> C interface is disabled. Timer I is cleared and does not run. This is the state assumed after a reset. If an I <sup>2</sup> C application wants to ignore the I <sup>2</sup> C at certain times, it should write SLAVEN, MASTRQ, and TIRUN all to zero.
All 0	1	The I <sup>2</sup> C interface is disabled. Timer I operates as a free-running time base. Use this mode only in non-I <sup>2</sup> C applications.
Any or all 1	0	The I <sup>2</sup> C interface is enabled. The 3 low-order bits of Timer I run for min-time generation, but the hi-order bits do not, so that there is no checking for I <sup>2</sup> C being "hung." This configuration can be used for very slow I <sup>2</sup> C operation.
Any or all 1	1	The I <sup>2</sup> C interface is enabled. Timer I runs during frames on the I <sup>2</sup> C, and is cleared by transitions on SCL, and by Start and Stop conditions. This is the normal state for I <sup>2</sup> C operation.

Values to be used in the CT1 and CT0 bits are shown in Table 5. To allow the I<sup>2</sup>C bus to run at the maximum rate for a particular oscillator frequency, compare the actual oscillator rate to the  $f_{osc}$  max column in the table. The value for CT1 and CT0 is found in the first line of the table where  $f_{osc}$  max is greater than or equal to the actual frequency.

The table also shows the osc/12 count for various settings of CT1/CT0. This allows calculation of the actual minimum high and low times for SCL as follows:

$$SCL \text{ min high/low time} = \frac{12 \cdot \text{count}}{osc} \text{ (in microseconds)}$$

For instance, at a 16MHz frequency, with CT1/CT0 set to 10, the minimum SCL high and low times will be 5.25 $\mu$ s

The table also shows the Timer I timeout period (given in machine cycles) for each CT1/CT0 combination. The timeout period varies because of the way in which minimum SCL high and low times are measured. When the I<sup>2</sup>C interface is operating, Timer I is preloaded at every SCL transition with a value dependent upon CT1/CT0. The preload value is chosen such that a minimum SCL high or low time has elapsed when Timer I reaches a count of 008 (the actual value preloaded into Timer I is 8 minus the osc/12 count).

**Table 5. CT1, CT0 Values**

CT1, CT0	OSC/12 COUNT	$f_{osc}$ MAX	TIMEOUT PERIOD
10	7	16.8MHz	1023 cycles
01	6	14.4MHz	1022 cycles
00	5	12.0MHz	1021 cycles
11	4	9.6MHz	1020 cycles

### I<sup>2</sup>C Register I2STA

Read only							
7	6	5	4	3	2	1	0
-	IDLE	XDATA	XACTV	MAKSTR	MAKSTP	XSTR	XSTP

This register is read only and reflects the internal status of the I<sup>2</sup>C hardware. IDLE, XSTR, and XSTP reflect the status of the like named bits in the I2CON register.

- XDATA** The content of the transmitter buffer.
- XACTV** Transmitter active.
- MAKSTR** This bit is high while the hardware is effecting a start condition.
- MAKSTP** This bit is high while the hardware is effecting a stop condition.
- XSTR** This bit is active while the hardware is effecting a repeated start condition.
- XSTP** This bit is active while the hardware is effecting a repeated stop condition.

### Interrupts

The interrupt structure is a five-source, one-level interrupt system. Interrupt sources

common to the 80C51 are the external interrupts (INT0, INT1) and the timer/counter interrupt (ET0). The I<sup>2</sup>C interrupt (EI2) and Timer I interrupt (ETI) are the other two interrupt sources. The interrupt sources are listed below in their order of polling sequence priority.

Upon interrupt or reset the program counter is loaded with specific values for the appropriate interrupt service routine in program memory. These values are:

Program Memory		
Event	Address	Priority
Reset	000	Highest
INT0	003	
Counter/Timer 0	00B	
INT1	013	
Timer I	01B	
I <sup>2</sup> C	023	Lowest

The interrupt enable register (IE) is used to individually enable or disable the five interrupt sources. Bit EA in the interrupt enable register can be used to globally enable or disable all interrupt sources. The interrupt enable register is described below. All other interrupt details are based on the 80C51 interrupt architecture.



## CMOS single-chip 8-bit microcontroller 83C751/87C751

## Interrupt Enable Register

EA	X	X	IE2	ETI	EX1	ETO	EX0
----	---	---	-----	-----	-----	-----	-----

## Symbol Position Function

EA	IE.7	Disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	Reserved
—	IE.5	Reserved
IE2	IE.4	Enables or disables the I <sup>2</sup> C interrupt. If IE2 = 0, the I <sup>2</sup> C interrupt is disabled.
ETI	IE.3	Enables or disables the Timer 1 overflow interrupt. If ETI = 0, the Timer 1 interrupt is disabled.
EX1	IE.2	Enables or disables external interrupt 1. If EX1 = 0, external interrupt 1 is disabled.
ETO	IE.1	Enables or disables the Timer 0 overflow interrupt. If ETO = 0, the Timer 0 interrupt is disabled.
EX0	IE.0	Enables or disables external interrupt 0. If EX0 = 0, external interrupt 0 is disabled.

1. Additional programming of the encryption key is inhibited.
2. Verification of the encryption key is inhibited.
3. Verification of the USER EPROM and the security bit levels may still be performed.
4. If the encryption key may be used, this security bit should be programmed by the user to prevent unauthorized access from reprogramming the encryption key to all logical zero bits. Such programming would provide data during a verify cycle that is the logical complement of the USER EPROM contents.
5. Security bit 2, the verify inhibit bit, prevents verification of both the USER EPROM array and the encryption key memory. The security bit levels may still be verified.

Initially be held high for at least two machine cycles. P0.1 (PGM) and P0.2 (Vpp) will be at V<sub>CC</sub> as a result of the RESET operation. At this point, these pins function as normal quasi-bidirectional I/O ports and the programming equipment may pull these lines low. However, prior to sending the 16-bit code on the RESET pin, the programming equipment should drive these pins high (V<sub>CC</sub>). The RESET pin may now be used as the serial data input for the data stream which places the 87C751 in the programming mode. Data bits are sampled during the clock high time and thus should only change during the time that the clock is low. Following transmission of the last data bit, the RESET pin should be held low.

Next the address information for the location to be programmed is placed on port 3 and ASSEL is used to perform the address multiplexing, as previously described. At this time, port 1 functions as an output. A high voltage V<sub>pp</sub> level is then applied to the V<sub>pp</sub> input (P0.2). This sets Port 1 as an input. The data to be programmed into the EPROM array is then placed on Port 1. This is followed by a series of programming pulses applied to the PGM pin (P0.1). These pulses are created by driving P0.1 low and then high. This pulse is repeated until a total of 32 programming pulses have occurred. At the conclusion of the last pulse, the PGM signal should remain high.

The V<sub>pp</sub> signal may now be driven to the V<sub>CC</sub> level, placing the 87C751 in the verify mode. (Port 1 is now used as an output port). After four machine cycles (48 clock periods), the contents of the addressed location in the EPROM array will appear on Port 1.

The next programming cycle may now be initiated by placing the address information at the inputs of the multiplexed buffers, driving the V<sub>pp</sub> pin to the V<sub>pp</sub> voltage level, providing the byte to be programmed to Port 1 and issuing the 32 programming pulses on the PGM pin, bringing V<sub>pp</sub> back down to the V<sub>CC</sub> level and verifying the byte.

## Programming Modes

The 87C751 has four programming features incorporated within the EPROM array. These include the USER EPROM for storage of the application's code, a 16-byte encryption key array and two security bits. Programming and verification of these four elements are selected by a combination of the serial data stream applied to the RESET pin and the voltage levels applied to port pins P0.1 and

## 87C751 PROGRAMMING CONSIDERATIONS

## EPROM Characteristics

The 87C751 is programmed by using a modified Quick-Pulse Programming algorithm similar to that used for devices such as the 87C621 and 87C61. It differs from these devices in that a serial data stream is used to place the 87C751 in the programming mode.

Figure 8 shows a block diagram of the programming configuration for the 87C751. Port pin P0.2 is used as the programming voltage supply output (V<sub>pp</sub> signal). Port pin P0.1 is used as the program (PGM) signal. This pin is used for the 32 programming pulses.

Port 3 is used as the address input for the data to be programmed and accepted both by high and low components of the eleven-bit address. Multiplexing of these address components is performed using the ASSEL input. The user should drive the ASSEL input high and then drive port 3 with the high order bits of the address. ASSEL should remain high for at least 18 clock cycles. ASSEL may then be driven low which latches the high order bits of the address internally. The high address should remain on port 3 for at least two clock cycles after ASSEL is driven low. Port 3 may then be driven with the low byte of the address. The low address will be internally latched 18 clock cycles later. The address will remain stable provided that the low byte placed on port 3 is held stable and ASSEL is kept low. Note: ASSEL needs to be pulsed high only to change the high byte of the address.

Port 1 is used as a bidirectional data bus during programming and verify operations. During programming mode, it accepts the byte to be programmed. During verify mode, it provides the contents of the EPROM location specified by the address which has been applied to Port 3.

The XTAL1 pin is the oscillator input and receives the master system clock. The clock should be between 1.5 and 5 MHz.

The RESET pin is used to accept the serial data stream that places the 87C751 into various programming modes. This pattern consists of a 16-bit code with the LSB code bit. Each bit is synchronized to the clock input X1.

## Programming Operation

Figure 9 and 10 show the timing diagrams for the program/verify cycle. RESET should



## CMOS single-chip 8-bit microcontroller 83C751/87C751

**87C751 PROGRAMMING CONSIDERATIONS****EPROM Characteristics**

The 87C751 is programmed by using a modified Quick-Pulse Programming algorithm similar to that used for devices such as the 87C451 and 87C51. It differs from these devices in that a serial data stream is used to place the 87C751 in the programming mode.

Figure 5 shows a block diagram of the programming configuration for the 87C751. Port pin P0.2 is used as the programming voltage supply input ( $V_{PP}$  signal). Port pin P0.1 is used as the program (PGM) signal. This pin is used for the 25 programming pulses.

Port 3 is used as the address input for the byte to be programmed and accepts both the high and low components of the eleven bit address. Multiplexing of these address components is performed using the ASEL input. The user should drive the ASEL input high and then drive port 3 with the high order bits of the address. ASEL should remain high for at least 13 clock cycles. ASEL may then be driven low which latches the high order bits of the address internally. The high address should remain on port 3 for at least two clock cycles after ASEL is driven low. Port 3 may then be driven with the low byte of the address. The low address will be internally stable 13 clock cycles later. The address will remain stable provided that the low byte placed on port 3 is held stable and ASEL is kept low. Note: ASEL needs to be pulsed high only to change the high byte of the address.

Port 1 is used as a bidirectional data bus during programming and verify operations. During programming mode, it accepts the byte to be programmed. During verify mode, it provides the contents of the EPROM location specified by the address which has been supplied to Port 3.

The XTAL1 pin is the oscillator input and receives the master system clock. This clock should be between 1.2 and 6MHz.

The RESET pin is used to accept the serial data stream that places the 87C751 into various programming modes. This pattern consists of a 10-bit code with the LSB sent first. Each bit is synchronized to the clock input, X1.

**Programming Operation**

Figures 6 and 7 show the timing diagrams for the program/verify cycle. RESET should

initially be held high for at least two machine cycles. P0.1 (PGM) and P0.2 ( $V_{PP}$ ) will be at  $V_{OH}$  as a result of the RESET operation. At this point, these pins function as normal quasi-bidirectional I/O ports and the programming equipment may pull these lines low. However, prior to sending the 10-bit code on the RESET pin, the programming equipment should drive these pins high ( $V_{IH}$ ). The RESET pin may now be used as the serial data input for the data stream which places the 87C751 in the programming mode. Data bits are sampled during the clock high time and thus should only change during the time that the clock is low. Following transmission of the last data bit, the RESET pin should be held low.

Next the address information for the location to be programmed is placed on port 3 and ASEL is used to perform the address multiplexing, as previously described. At this time, port 1 functions as an output.

A high voltage  $V_{PP}$  level is then applied to the  $V_{PP}$  input (P0.2). (This sets Port 1 as an input port). The data to be programmed into the EPROM array is then placed on Port 1. This is followed by a series of programming pulses applied to the PGM/ pin (P0.1). These pulses are created by driving P0.1 low and then high. This pulse is repeated until a total of 25 programming pulses have occurred. At the conclusion of the last pulse, the PGM/ signal should remain high.

The  $V_{PP}$  signal may now be driven to the  $V_{OH}$  level, placing the 87C751 in the verify mode. (Port 1 is now used as an output port). After four machine cycles (48 clock periods), the contents of the addressed location in the EPROM array will appear on Port 1.

The next programming cycle may now be initiated by placing the address information at the inputs of the multiplexed buffers, driving the  $V_{PP}$  pin to the  $V_{PP}$  voltage level, providing the byte to be programmed to Port 1 and issuing the 26 programming pulses on the PGM/ pin, bringing  $V_{PP}$  back down to the  $V_C$  level and verifying the byte.

**Programming Modes**

The 87C751 has four programming features incorporated within its EPROM array. These include the USER EPROM for storage of the application's code, a 16-byte encryption key array and two security bits. Programming and verification of these four elements are selected by a combination of the serial data stream applied to the RESET pin and the voltage levels applied to port pins P0.1 and

P0.2. The various combinations are shown in Table 6.

**Encryption Key Table**

The 87C751 includes a 16-byte EPROM array that is programmable by the end user. The contents of this array can then be used to encrypt the program memory contents during a program memory verify operation. When a program memory verify operation is performed, the contents of the program memory location is XNOR'ed with one of the bytes in the 16-byte encryption table. The resulting data pattern is then provided to port 1 as the verify data. The encryption mechanism can be disable, in essence, by leaving the bytes in the encryption table in their erased state (FFH) since the XNOR product of a bit with a logical one will result in the original bit. The encryption bytes are mapped with the code memory in 16-byte groups; the first byte in code memory will be encrypted with the first byte in the encryption table; the second byte in code memory will be encrypted with the second byte in the encryption table and so forth up to and including the 16th byte. The encryption repeats in 16-byte groups; the 17th byte in the code memory will be encrypted with the first byte in the encryption table, and so forth.

**Security Bits**

Two security bits, security bit 1 and security bit 2, are provided to limit access to the USER EPROM and encryption key arrays. Security bit 1 is the program inhibit bit, and once programmed performs the following functions:

1. Additional programming of the USER EPROM is inhibited.
2. Additional programming of the encryption key is inhibited.
3. Verification of the encryption key is inhibited.
4. Verification of the USER EPROM and the security bit levels may still be performed.

(If the encryption key array is being used, this security bit should be programmed by the user to prevent unauthorized parties from reprogramming the encryption key to all logical zero bits. Such programming would provide data during a verify cycle that is the logical complement of the USER EPROM contents).

Security bit 2, the verify inhibit bit, prevents verification of both the USER EPROM array and the encryption key arrays. The security bit levels may still be verified.

## CMOS single-chip 8-bit microcontroller 83C751/87C751

### Programming and Verifying Security Bits

Security bits are programmed employing the same techniques used to program the USER EPROM and KEY arrays using serial data streams and logic levels on port pins indicated in Table 6. When programming either security bit, it is not necessary to provide address or data information to the 87C751 on ports 1 and 3.

Verification occurs in a similar manner using the RESET serial stream shown in Table 6. Port 3 is not required to be driven and the results of the verify operation will appear on ports 1.6 and 1.7.

Ports 1.7 contains the security bit 1 data and is a logical one if programmed and a logical zero if erased. Likewise, P1.6 contains the security bit 2 data and is a logical one if programmed and a logical zero if erased.

### Erasure Characteristics

Erasure of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent

erasure. For this and secondary effects, it is recommended that an opaque label be placed over the window. For elevated temperature or environments where solvents are being used, apply Kapton tape Flourless part number 2345-5 or equivalent.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 angstroms) to an integrated dose of at least 15W-s/cm<sup>2</sup>. Exposing the EPROM to an ultraviolet lamp of 12,000µW/cm<sup>2</sup> rating for 20 to 39 minutes, at a distance of about 1 inch, should be sufficient.

Erasure leaves the array in an all 1s state.

Table 6. Implementing Program/Verify Modes

OPERATION	SERIAL CODE	P0.1 (PGM/)	P0.2 (V <sub>PP</sub> )
Program user EPROM	296H	—*	V <sub>PP</sub>
Verify user EPROM	296H	V <sub>IH</sub>	V <sub>IH</sub>
Program key EPROM	292H	—*	V <sub>PP</sub>
Verify key EPROM	292H	V <sub>IH</sub>	V <sub>IH</sub>
Program security bit 1	29AH	—*	V <sub>PP</sub>
Program security bit 2	298H	—*	V <sub>PP</sub>
Verify security bits	29AH	V <sub>IH</sub>	V <sub>IH</sub>

#### NOTE:

\* Pulsed from V<sub>IH</sub> to V<sub>IL</sub> and returned to V<sub>IH</sub>.

### EPROM PROGRAMMING AND VERIFICATION

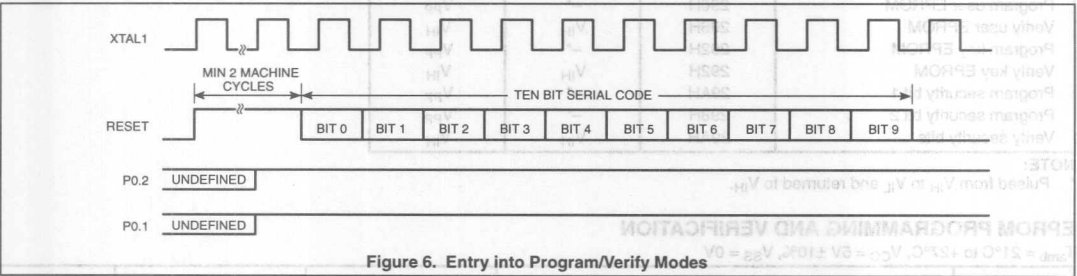
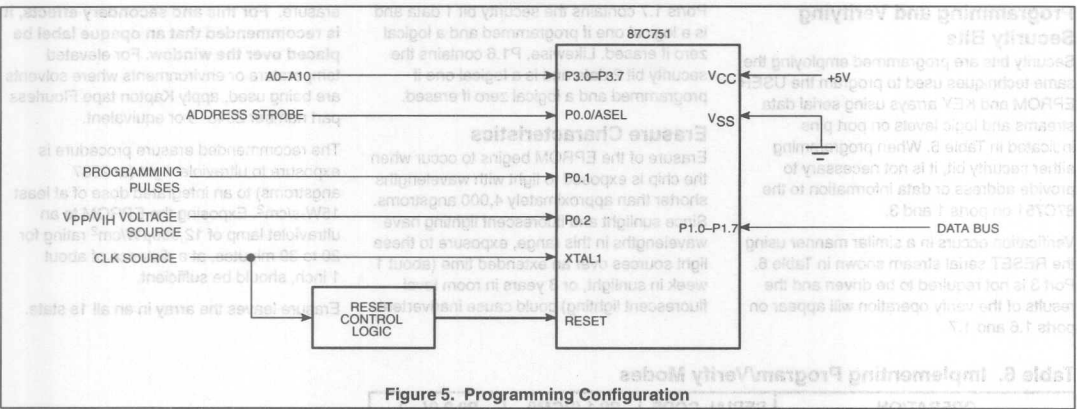
T<sub>amb</sub> = 21°C to +27°C, V<sub>CC</sub> = 5V ±10%, V<sub>SS</sub> = 0V

SYMBOL	PARAMETER	MIN	MAX	UNIT
1/f <sub>CLCL</sub>	Oscillator/clock frequency	1.2	6	MHz
t <sub>AVGL</sub> <sup>1</sup>	Address setup to P0.1 (PROG-) low	10µs + 24t <sub>CLCL</sub>		
t <sub>GHAX</sub>	Address hold after P0.1 (PROG-) high	48t <sub>CLCL</sub>		
t <sub>DVGL</sub>	Data setup to P0.1 (PROG-) low	38t <sub>CLCL</sub>		
t <sub>GHDX</sub>	Data hold after P0.1 (PROG-) high	36t <sub>CLCL</sub>		
t <sub>SHGL</sub>	V <sub>PP</sub> setup to P0.1 (PROG-) low	10		µs
t <sub>GHSL</sub>	V <sub>PP</sub> hold after P0.1 (PROG-)	10		µs
t <sub>GLGH</sub>	P0.1 (PROG-) width	90	110	µs
t <sub>AVQV</sub> <sup>2</sup>	V <sub>PP</sub> low (V <sub>CC</sub> ) to data valid		48t <sub>CLCL</sub>	
t <sub>GHGL</sub>	P0.1 (PROG-) high to P0.1 (PROG-) low	10		µs
t <sub>MASEL</sub>	ASEL high time	13t <sub>CLCL</sub>		
t <sub>HAHLD</sub>	Address hold time	2t <sub>CLCL</sub>		
t <sub>HASET</sub>	Address setup to ASEL	13t <sub>CLCL</sub>		
t <sub>ADSTA</sub>	Low address to valid data		48t <sub>CLCL</sub>	

#### NOTES:

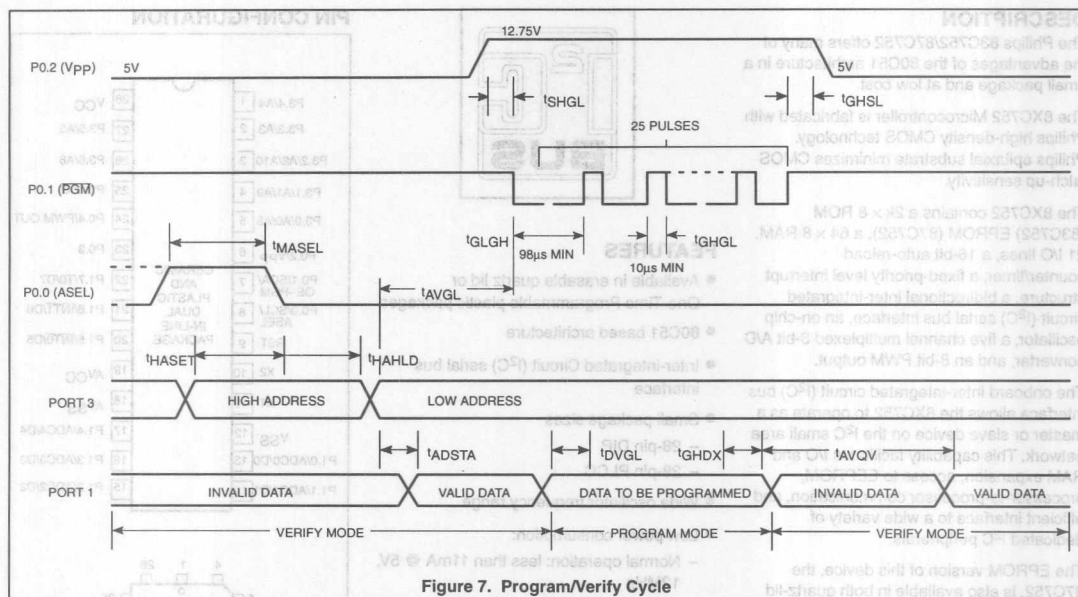
- Address should be valid at least 24t<sub>CLCL</sub> before the rising edge of P0.2 (V<sub>PP</sub>).
- For a pure verify mode, i.e., no program mode in between, t<sub>AVQV</sub> is 14t<sub>CLCL</sub> maximum.

CMOS single-chip 8-bit microcontroller 83C751/87C751



Symbol	Parameter	Unit	Value
$f_{osc}$	Oscillation frequency	Hz	1.2
$t_{addr}$	Address setup to P0.1 (PROG-) low	ns	10
$t_{addr}$	Address hold after P0.1 (PROG-) high	ns	10
$t_{data}$	Data setup to P0.1 (PROG-) low	ns	10
$t_{data}$	Data hold after P0.1 (PROG-) high	ns	10
$t_{vpp}$	Vpp setup to P0.1 (PROG-) low	ns	10
$t_{vpp}$	Vpp hold after P0.1 (PROG-) high	ns	10
$t_{p0.1}$	P0.1 (PROG-) width	ns	10
$t_{vpp}$	Vpp low (Vcc) to data valid	ns	10
$t_{p0.1}$	P0.1 (PROG-) high to P0.1 (PROG-) low	ns	10
$t_{ase}$	ASEL high time	ns	10
$t_{ah}$	Address hold time	ns	10
$t_{as}$	Address setup to ASEL	ns	10
$t_{as}$	Low address to valid data	ns	10

## CMOS single-chip 8-bit microcontroller 83C751/87C751



Purchase of Philips' I<sup>2</sup>C components conveys a license under the Philips' I<sup>2</sup>C patent to use the components in the I<sup>2</sup>C-system provided the system conforms to the I<sup>2</sup>C specifications defined by Philips.

# CMOS single-chip 8-bit microcontroller 83C752/87C752

CMOS single-chip 8-bit microcontroller with A/D, PWM

## DESCRIPTION

The Philips 83C752/87C752 offers many of the advantages of the 80C51 architecture in a small package and at low cost.

The 83C752 Microcontroller is fabricated with Philips high-density CMOS technology. Philips epitaxial substrate minimizes CMOS latch-up sensitivity.

The 83C752 contains a  $2k \times 8$  ROM (83C752) EPROM (87C752), a  $64 \times 8$  RAM, 21 I/O lines, a 16-bit auto-reload counter/timer, a fixed-priority level interrupt structure, a bidirectional inter-integrated circuit (I<sup>2</sup>C) serial bus interface, an on-chip oscillator, a five channel multiplexed 8-bit A/D converter, and an 8-bit PWM output.

The onboard inter-integrated circuit (I<sup>2</sup>C) bus interface allows the 83C752 to operate as a master or slave device on the I<sup>2</sup>C small area network. This capability facilitates I/O and RAM expansion, access to EEPROM, processor-to-processor communication, and efficient interface to a wide variety of dedicated I<sup>2</sup>C peripherals.

The EPROM version of this device, the 87C752, is also available in both quartz-lid erasable and plastic one-time programmable (OTP) packages. Once the array has been programmed, it is functionally equivalent to the masked ROM 83C752. Thus, unless explicitly stated otherwise, all references made to the 83C752 apply equally to the 87C752.

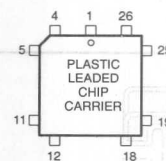
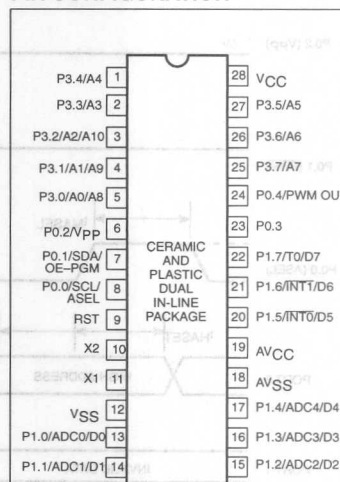
The 83C752 supports two power reduction modes of operation referred to as the idle mode and the power-down mode.



## FEATURES

- Available in erasable quartz lid or One-Time Programmable plastic packages
- 80C51 based architecture
- Inter-integrated Circuit (I<sup>2</sup>C) serial bus interface
- Small package sizes
  - 28-pin DIP
  - 28-pin PLCC
- Wide oscillator frequency range
- Low power consumption:
  - Normal operation: less than 11mA @ 5V, 12MHz
  - Idle mode
  - Power-down mode
- $2k \times 8$  ROM (83C752) EPROM (87C752)
- $64 \times 8$  RAM
- 16-bit auto reloadable counter/timer
- 5-channel 8-bit A/D converter
- 8-bit PWM output/timer
- Fixed-rate timer
- Boolean processor
- CMOS and TTL compatible
- Well suited for logic replacement, consumer and industrial applications

## PIN CONFIGURATION



Pin	Function	Pin	Function
1	P3.4/A4	15	P1.2/ADC2/D2
2	P3.3/A3	16	P1.3/ADC3/D3
3	P3.2/A2/A10	17	P1.4/ADC4/D4
4	P3.1/A1/A9	18	AVss
5	P3.0/A0/A8	19	AVcc
6	P0.2/Vpp	20	P1.5/INT0/D5
7	P0.1/SDA/OE-PGM	21	P1.6/INT1/D6
8	P0.0/SCL/ASEL	22	P1.7/T0/D7
9	RST	23	P0.3
10	X2	24	P0.4/PWM OUT
11	X1	25	P3.7/A7
12	Vss	26	P3.6/A6
13	P1.0/ADC0/D0	27	P3.5/A5
14	P1.1/ADC1/D1	28	Vcc



## CMOS single-chip 8-bit microcontroller 83C752/87C752

## PART NUMBER SELECTION

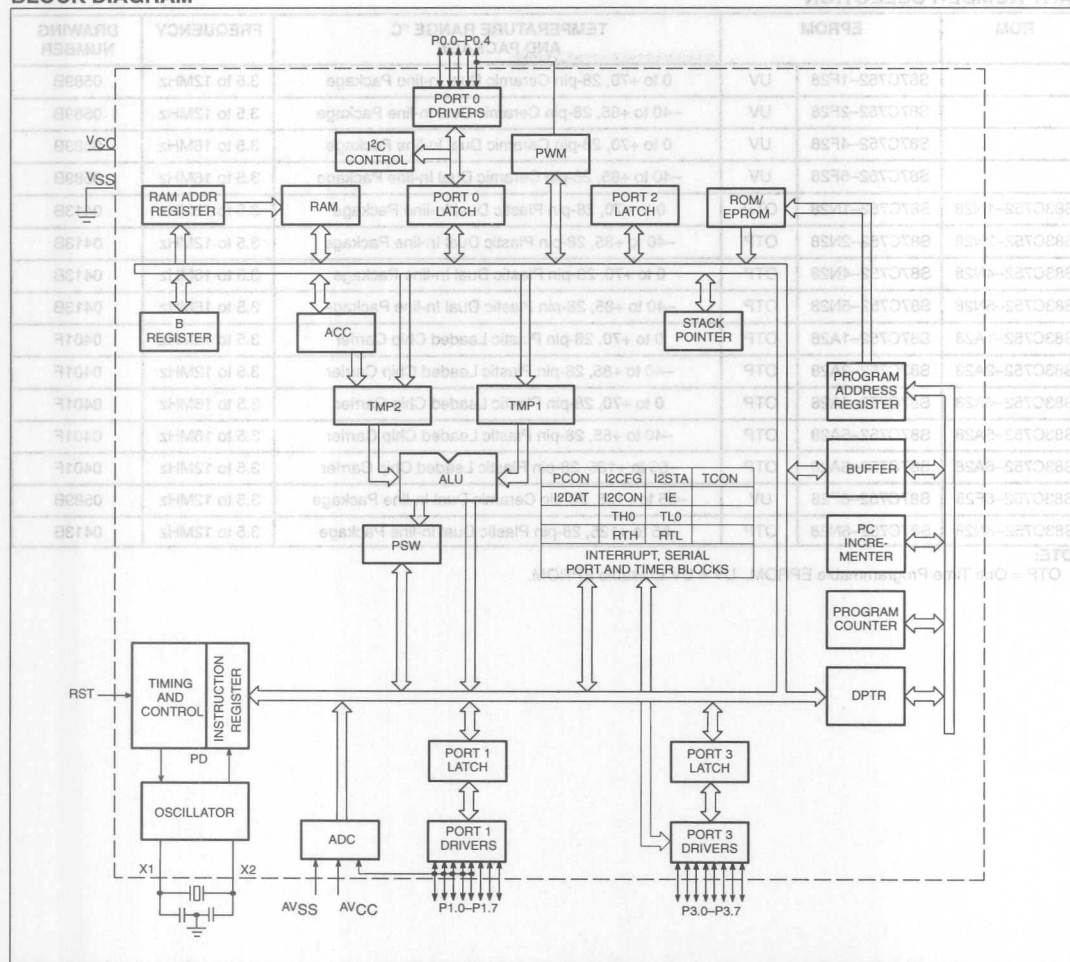
ROM	EPROM		TEMPERATURE RANGE °C AND PACKAGE	FREQUENCY	DRAWING NUMBER
	S87C752-1F28	UV	0 to +70, 28-pin Ceramic Dual In-line Package	3.5 to 12MHz	0589B
	S87C752-2F28	UV	-40 to +85, 28-pin Ceramic Dual In-line Package	3.5 to 12MHz	0589B
	S87C752-4F28	UV	0 to +70, 28-pin Ceramic Dual In-line Package	3.5 to 16MHz	0589B
	S87C752-5F28	UV	-40 to +85, 28-pin Ceramic Dual In-line Package	3.5 to 16MHz	0589B
S83C752-1N28	S87C752-1N28	OTP	0 to +70, 28-pin Plastic Dual In-line Package	3.5 to 12MHz	0413B
S83C752-2N28	S87C752-2N28	OTP	-40 to +85, 28-pin Plastic Dual In-line Package	3.5 to 12MHz	0413B
S83C752-4N28	S87C752-4N28	OTP	0 to +70, 28-pin Plastic Dual In-line Package	3.5 to 16MHz	0413B
S83C752-5N28	S87C752-5N28	OTP	-40 to +85, 28-pin Plastic Dual In-line Package	3.5 to 16MHz	0413B
S83C752-1A28	S87C752-1A28	OTP	0 to +70, 28-pin Plastic Leaded Chip Carrier	3.5 to 12MHz	0401F
S83C752-2A28	S87C752-2A28	OTP	-40 to +85, 28-pin Plastic Leaded Chip Carrier	3.5 to 12MHz	0401F
S83C752-4A28	S87C752-4A28	OTP	0 to +70, 28-pin Plastic Leaded Chip Carrier	3.5 to 16MHz	0401F
S83C752-5A28	S87C752-5A28	OTP	-40 to +85, 28-pin Plastic Leaded Chip Carrier	3.5 to 16MHz	0401F
S83C752-6A28	S87C752-6A28	OTP	-55 to +125, 28-pin Plastic Leaded Chip Carrier	3.5 to 12MHz	0401F
S83C752-6F28	S87C752-6F28	UV	-55 to +125, 28-pin Ceramic Dual In-line Package	3.5 to 12MHz	0589B
S83C752-6N28	S87C752-6N28	OTP	-55 to +125, 28-pin Plastic Dual In-line Package	3.5 to 12MHz	0413B

**NOTE:**

1. OTP = One Time Programmable EPROM. UV = UV Erasable EPROM.

## CMOS single-chip 8-bit microcontroller 83C752/87C752

## BLOCK DIAGRAM



## CMOS single-chip 8-bit microcontroller 83C752/87C752

## PIN DESCRIPTION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
V <sub>SS</sub>	12	I	<b>Circuit Ground Potential.</b>
V <sub>CC</sub>	28	I	<b>Supply voltage during normal, idle, and power-down operation.</b>
P0.0–P0.4	8–6 23, 24	I/O	<b>Port 0:</b> Port 0 is a 5-bit bidirectional port. Port 0.0–P0.2 are open drain. Port 0.0–P0.2 pins that have 1s written to them float, and in that state can be used as high-impedance inputs. P0.3–P0.4 are bidirectional I/O port pins with internal pull-ups. Port 0 also serves as the serial I <sup>2</sup> C interface. When this feature is activated by software, SCL and SDA are driven low in accordance with the I <sup>2</sup> C protocol. These pins are driven low if the port register bit is written with a 0 or if the I <sup>2</sup> C subsystem presents a 0. The state of the pin can always be read from the port register by the program. Port 0.3 and 0.4 have internal pull-ups that function identically to port 3. Pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs.  To comply with the I <sup>2</sup> C specification, P0.0 and P0.1 are open drain bidirectional I/O pins with the electrical characteristics listed in the tables that follow. While these differ from "standard TTL" characteristics, they are close enough for the pins to still be used as general-purpose I/O in non-I <sup>2</sup> C applications.
	6	I	<b>V<sub>PP</sub> (P0.2)</b> – Programming voltage input.
	7	I	<b>OE/PGM (P0.1)</b> – Input which specifies verify mode (output enable) or the program mode. OE/PGM = 1 output enabled (verify mode). OE/PGM = 0 program mode.
	8	I	<b>ASEL (P0.0)</b> – Input which indicates which bits of the EPROM address are applied to port 3. ASEL = 0 low address byte available on port 3. ASEL = 1 high address byte available on port 3 (only the three least significant bits are used).
P1.0–P1.7	13–17, 20–22	I/O	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. P0.3–P0.4 pins are bidirectional I/O port pins with internal pull-ups. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 1 also serves the special function features of the SC80C51 family as listed below: <b>INT0 (P1.5):</b> External interrupt. <b>INT1 (P1.6):</b> External interrupt. <b>T0 (P1.7):</b> Timer 0 external input. <b>ADC0 (P1.0)–ADC4 (P1.4):</b> Port 1 also functions as the inputs to the five channel multiplexed A/D converter. These pins can be used as outputs only if the A/D function has been disabled. These pins can be used as inputs while the A/D converter is enabled.  Port 1 serves to output the addressed EPROM contents in the verify mode and accepts as inputs the value to program into the selected address during the program mode.
P3.0–P3.7	5–1, 27–25	I/O	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 3 also functions as the address input for the EPROM memory location to be programmed (or verified). The 11-bit address is multiplexed into this port as specified by P0.0/ASEL.
RST	9	I	<b>Reset:</b> A high on this pin for two machine cycles while the oscillator is running resets the device. An internal diffused resistor to V <sub>SS</sub> permits a power-on RESET using only an external capacitor to V <sub>CC</sub> . After the device is reset, a 10-bit serial sequence, sent LSB first, applied to RESET, places the device in the programming state allowing programming address, data and V <sub>PP</sub> to be applied for programming or verification purposes. The RESET serial sequence must be synchronized with the X1 input.
X1	11	I	<b>Crystal 1:</b> Input to the inverting oscillator amplifier and input to the internal clock generator circuits. X1 also serves as the clock to strobe in a serial bit stream into RESET to place the device in the programming state.
X2	10	O	<b>Crystal 2:</b> Output from the inverting oscillator amplifier.
AV <sub>CC</sub> <sup>1</sup>	19	I	<b>Analog supply voltage and reference input.</b>
AV <sub>SS</sub> <sup>1</sup>	18	I	<b>Analog supply and reference ground.</b>

## NOTE:

1. AV<sub>SS</sub> (reference ground) must be connected to 0V (ground). AV<sub>CC</sub> (reference input) cannot differ from V<sub>CC</sub> by more than  $\pm 0.2V$ , and must be in the range 4.5V to 5.5V.

# CMOS single-chip 8-bit microcontroller 83C752/87C752

## OSCILLATOR

### CHARACTERISTICS

X1 and X2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator.

To drive the device from an external clock source, X1 should be driven while X2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

## IDLE MODE

The 83C752 includes the 80C51 power-down and idle mode features. In idle mode, the CPU puts itself to sleep while all of the on-chip peripherals except the A/D and PWM stay active. The functions that continue to run while in the idle mode are Timer 0, the I<sup>2</sup>C interface including Timer 1, and the interrupts. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset. Upon powering-up the circuit, or exiting from idle mode, sufficient time must be allowed for stabilization of the internal analog reference voltages before an A/D conversion is started.

## Special Function Registers

The special function registers (directly addressable only) contain all of the 83C751 registers except the program counter and the four register banks. Most of the 21 special function registers are used to control the on-chip peripheral hardware. Other registers include arithmetic registers (ACC, B, PSW), stack pointer (SP) and data pointer registers (DPH, DPL). Nine of the SFRs are bit addressable.

## Data Pointer

The data pointer (DPTR) consists of a high byte (DPH) and a low byte (DPL). In the 80C51 this register allows the access of external data memory using the MOVX instruction. Since the 83C752 does not support MOVX or external memory accesses, this register is generally used as a 16-bit offset pointer of the accumulator in a MOVC instruction. DPTR may also be manipulated as two independent 8-bit registers.

## POWER-DOWN MODE

In the power-down mode, the oscillator is stopped and the instruction to invoke power-down is the last instruction executed. Only the contents of the on-chip RAM are preserved. A hardware reset is the only way to terminate the power-down mode. The control bits for the reduced power modes are in the special function register PCON.

**Table 1. External Pin Status During Idle and Power-Down Modes**

MODE	Port 0*	Port 1	Port 2
Idle	Data	Data	Data
Power-down	Data	Data	Data

\* Except for PWM output (P0.4).

## DIFFERENCES BETWEEN THE 83C752 AND THE 80C51

### Program Memory

On the 83C752, program memory is 2048 bytes long and is not externally expandable, so the 80C51 instructions MOVX, LJMP, and LCALL are not implemented. If these instructions are executed, the appropriate number of instruction cycles will take place along with external fetches; however, no operation will take place. The LJMP may not respond to all program address bits. The only fixed locations in program memory are the addresses at which execution is taken up in response to reset and interrupts, which are as follows:

Event	Address
Reset	000
External INT0	003
Counter/timer 0	00B
External INT1	013
Timer 1	01B
I <sup>2</sup> C serial	023
ADC	02B
PWM	033

## Memory Organization

The 83C752 manipulates operands in three memory address spaces. The first is the program memory space which contains program instructions as well as constants such as look-up tables. The program memory space contains 2k bytes in the 83C752.

The second memory space is the data memory array which has a logical address space of 128 bytes. However, only the first 64 (0 to 3FH) are implemented in the 83C752.

The third memory space is the special function register array having a 128-byte address space (80H to FFH). Only selected locations in this memory space are used (see

Table 1). Note that the architecture of these memory spaces (internal program memory, internal data memory, and special function registers) is identical to the 80C51, and the 83C752 varies only in the amount of memory physically implemented.

The 83C752 does not directly address any external data or program memory spaces. For this reason, the MOVX instructions in the 80C51 instruction set are not implemented in the 83C752, nor are the alternate I/O pin functions RD and WR.

## I/O Ports

The I/O pins provided by the 83C752 consist of port 0, port 1, and port 3.

### Port 0

Port 0 is a 5-bit bidirectional I/O port and includes alternate functions on some pins of this port. Pins P0.3 and P0.4 are provided with internal pullups while the remaining pins (P0.0, P0.1, and P0.2) have open drain output structures. The alternate functions for port 0 are:

P0.0	SCL — the I <sup>2</sup> C bus clock
P0.1	SDA — the I <sup>2</sup> C bus data
P0.4	PWM — the PWM output

If the alternate functions, I<sup>2</sup>C and PWM, are not being used, then these pins may be used as I/O ports.

### Port 1

Port 1 is an 8-bit bidirectional I/O port whose structure is identical to the 80C51, but also includes alternate input functions on all pins. The alternate pin functions for port 1 are:

P1.0-P1.4	ADC0-ADC4 - A/D converter analog inputs
P1.5	INT0 - external interrupt 0 input
P1.6	INT1 - external interrupt 1 input
P1.7	T0 - timer 0 external input

If the alternate functions INT0, INT1, or T0 are not being used, these pins may be used as standard I/O ports, provided that the A/D is disabled. It is necessary to connect AV<sub>CC</sub> and AV<sub>SS</sub> to V<sub>CC</sub> and V<sub>SS</sub>, respectively, in order to use these pins as standard I/O pins. When the A/D converter is enabled, the analog channel connected to the A/D may not be used as a digital input; however, the remaining analog inputs may be used as digital inputs. They may not be used as digital outputs. While the A/D is enabled, the analog inputs are floating.

### Port 3

Port 3 is an 8-bit bidirectional I/O port whose structure is identical to the 80C51. Note that the alternate functions associated with port 3 of the 80C51 have been moved to port 1 of the 83C752 (as applicable). See Figure 1 for port bit configurations.

## CMOS single-chip 8-bit microcontroller 83C752/87C752

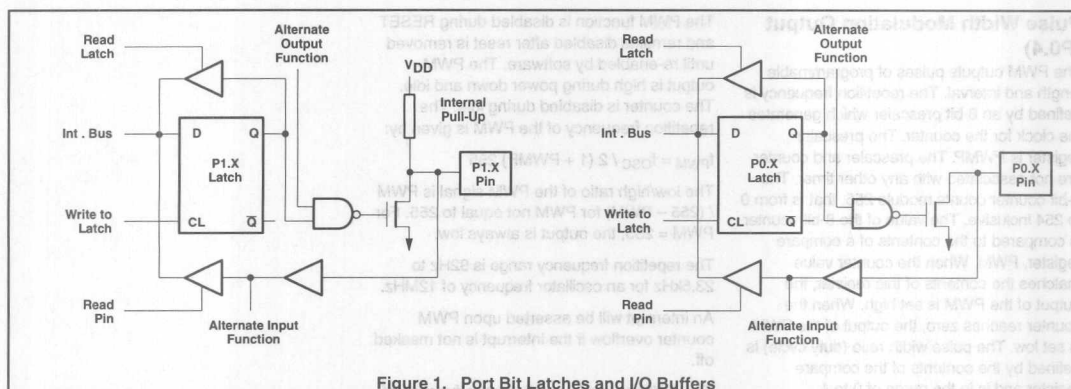


Figure 1. Port Bit Latches and I/O Buffers

**Counter/Timer Subsystem**

The 8XC752 has one counter/timer called timer/counter 0. Its operation is similar to mode 2 operation on the 80C51, but is extended to 16 bits with 16 bits of autoloop. The controls for this counter are centralized in a single register called TCON.

A watchdog timer, called Timer 1, is for use with the I<sup>2</sup>C subsystem. In I<sup>2</sup>C applications, this timer is dedicated to time-generation and bus monitoring of the I<sup>2</sup>C. In non-I<sup>2</sup>C applications, it is available for use as a fixed time-base.

**Interrupt Subsystem—Fixed Priority**

The IP register and the 2-level interrupt system of the 80C51 are eliminated. The interrupt structure is a seven-source, one-level interrupt system similar to the 8XC751. Simultaneous interrupt conditions are resolved by a single-level, fixed priority as follows:

Highest priority:	Pin INT0
	Counter/timer flag 0
	Pin INT1
	PWM
	Timer 1
	Serial I <sup>2</sup> C
Lowest priority:	ADC

The vector addresses are as follows:

Source	Vector Address
INT0	0003H
TF0	000BH
INT1	0013H
TIMER 1	001BH
SIO	0023H
ADC	002BH
PWM	0033H

**Interrupt Control Registers**

The 80C51 interrupt enable register is modified to take into account the different interrupt sources of the 8XC752.

**Interrupt Enable Register**

MSB	EA	EAD	ETI	ES	EPWM	EX1	ET0	EX0	LSB
-----	----	-----	-----	----	------	-----	-----	-----	-----

Position	Symbol	Function
IE.7	EA	Global interrupt disable when EA = 0
IE.6	EAD	A/D conversion complete
IE.5	ETI	Timer 1
IE.4	ES	I <sup>2</sup> C serial port
IE.3	EPWM	PWM counter overflow
IE.2	EX1	External interrupt 1
IE.1	ET0	Timer 0 overflow
IE.0	EX0	External interrupt 0

**Serial Communications**

The 8XC752 contains an I<sup>2</sup>C serial communications port instead of the 80C51 UART. The I<sup>2</sup>C serial port is a single bit hardware interface with all of the hardware necessary to support multimaster and slave operations. Also included are receiver digital filters and timer (timer 1) for communication watch-dog purposes. The I<sup>2</sup>C serial port is controlled through four special function

registers; I<sup>2</sup>C control, I<sup>2</sup>C data, I<sup>2</sup>C status, and I<sup>2</sup>C configuration.

The I<sup>2</sup>C bus uses two wires (SDA and SCL) to transfer information between devices connected to the bus. The main technical features of the bus are:

- Bidirectional data transfer between masters and slaves
- Serial addressing of slaves
- Acknowledgment after each transferred byte
- Multimaster bus
- Arbitration between simultaneously transmitting master without corruption of serial data on bus
- With 82B715, communication distance is extended to excess of 100 feet (30M)

A large family of I<sup>2</sup>C compatible ICs is available. See the I<sup>2</sup>C section for more details on the bus and available ICs.

The 83C752 I<sup>2</sup>C subsystem includes hardware to simplify the software required to drive the I<sup>2</sup>C bus. This circuitry is the same as that on the 83C751. (See the 83C751 section for a detailed discussion of this subsystem).



## CMOS single-chip 8-bit microcontroller 83C752/87C752

### Pulse Width Modulation Output (P0.4)

The PWM outputs pulses of programmable length and interval. The repetition frequency is defined by an 8-bit prescaler which generates the clock for the counter. The prescaler register is PWMP. The prescaler and counter are not associated with any other timer. The 8-bit counter counts modulo 255, that is from 0 to 254 inclusive. The value of the 8-bit counter is compared to the contents of a compare register, PWM. When the counter value matches the contents of this register, the output of the PWM is set high. When the counter reaches zero, the output of the PWM is set low. The pulse width ratio (duty cycle) is defined by the contents of the compare register and is in the range of 0 to 1. programmed in increments of 1/255. The PWM output can be set to be continuously high by loading the compare register with 0 and the output can be set to be continuously low by loading the compare register with 255. The PWM output is enabled by a bit in a special function register, PWENA. When enabled, the pin output is driven with a fully active pull-up. That is, when the output is high, a strong pull-up is continuously applied. When disabled, the pin functions as a normal bidirectional I/O pin, however, the counter remains active.

The PWM function is disabled during RESET and remains disabled after reset is removed until re-enabled by software. The PWM output is high during power down and idle. The counter is disabled during idle. The repetition frequency of the PWM is given by:

$$f_{PWM} = f_{OSC} / 2 (1 + PWMP) \quad 255$$

The low/high ratio of the PWM signal is  $\text{PWM} / (255 - \text{PWM})$  for PWM not equal to 255. For PWM = 255, the output is always low.

The repetition frequency range is 92Hz to 23.5kHz for an oscillator frequency of 12MHz.

An interrupt will be asserted upon PWM counter overflow if the interrupt is not masked off

The PWM output is an alternative function of P0.4. In order to use this port as a bidirectional I/O port, the PWM output must be disabled by clearing the enable/disable bit in PWENA. In this case, the PWM subsystem can be used as an interval timer by enabling the PWM interrupt.

## CMOS single-chip 8-bit microcontroller 83C752/87C752

Table 1. 83C752 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB							LSB	
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
ADAT#	A/D result	84H									00H
ADCON#	A/D control	A0H	-	-	ENADC	ADCI	ADCS	AADR2	AADR1	AADR0	C0H
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
DPTR:	Data pointer										
	(2 bytes)										
DPL	Data pointer low	82H									00H
DPH	Data pointer high	83H									00H
I <sup>2</sup> CFG#	I <sup>2</sup> C configuration	D8H/RD WR	DF	DE	DD	DC	DB	DA	D9	D8	0000xx00B
			SLAVEN	MASTRQ	0	TIRUN	-	-	CT1	CT0	
			SLAVEN	MASTRQ	CLRTI	TIRUN	-	-	CT1	CT0	
			9F	9E	9D	9C	9B	9A	99	98	
I <sup>2</sup> CON#	I <sup>2</sup> C control	98H/RD WR	RDAT	ATN	DRDY	ARL	STR	STP	MASTER	-	81H
			CXA	IDLE	CDR	CARL	CSTR	CSTP	XSTR	XSTP	
I <sup>2</sup> DAT#	I <sup>2</sup> C data	99H/RD WR	RDAT	0	0	0	0	0	0	0	80H
			XDAT	X	X	X	X	X	X	X	
			FF	FE	FD	FC	FB	FA	F9	F8	
I <sup>2</sup> STA#	I <sup>2</sup> C status	F8H	-	IDLE	XDATA	XACTV	MAKSTR	MAKSTP	XSTR	XSTP	x0100000B
			AF	AE	AD	AC	AB	AA	A9	A8	
IE#	Interrupt enable	A8H	EA	EAD	ETI	ES	EPWM	EX1	ET0	EX0	00H
			-	-	-	84	83	82	81	80	xxx11111B
P0#	Port 0	80H	-	-	-	PWM0	-	-	SDA	SCL	
			97	96	95	94	93	92	91	90	FFH
P1#	Port 1	90H	T0	INT1	INT0	ADC4	ADC3	ADC2	ADC1	ADC0	
P3#	Port 3	B0H	B7	B6	B5	B4	B3	B2	B1	B0	FFH
PCON#	Power control	87H	-	-	-	-	-	-	PD	IDL	xxxx0000B
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program status word	D0H	CY	AC	F0	RS1	RS0	OV	-	P	00H
PWCM#	PWM compare	8EH									xxxxxxxB
PWENA#	PWM enable	FEH	-	-	-	-	-	-	-	PWE	FEH
PWMP#	PWM prescaler	8FH									00H
RTL#	Timer low reload	8BH									00H
RTH#	Timer high reload	8DH									00H
SP	Stack pointer	81H									07H
TL#	Timer low	8AH									00H
TH#	Timer high	8CH									00H
			8F	8E	8D	8C	8B	8A	89	88	
TCON#	Timer control	88H	GATE	C/T	TF	TR	IE0	IT0	IE1	IT1	00H

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.

## CMOS single-chip 8-bit microcontroller 83C752/87C752

## Special Function Register

## Addresses

Special function registers for the 83C752 are identical to those of the 80C51, except for the changes listed below:

80C51 special function registers not present in the 83C752 are TMOD (89), P2 (A0) and IP (B8). The 80C51 registers TH1, TL1, SCON, and SBUF are replaced with the 83C752 registers RTH, RTL, I2CON, and I2DAT, respectively. Additional special function registers are I2CFG (D8) and I2STA (FB), ADCON (A0), ADAT (84), PWM (8E), PWMP (8F), and PWENA (FE). See Table 2.

## A/D Converter

The analog input circuitry consists of a 5-input analog multiplexer and an A to D converter with 8-bit resolution. The conversion takes 40 machine cycles, i.e., 40µs at 12MHz oscillator frequency. The A/D converter is controlled using the ADCON control register. Input channels are selected by the analog multiplexer through ADCON register bits 0–2.

The 83C752 contains a five-channel multiplexed 8-bit A/D converter. The conversion requires 40 machine cycles (40µs at 12MHz oscillator frequency).

The A/D converter is controlled by the A/D control register, ADCON. Input channels are selected by the analog multiplexer by bits ADCON.0 through ADCON.2. The ADCON register is not bit addressable.

## ADCON Register

MSB		LSB				
X	X	ENADC	ADCI	ADCS	AADR2	AADR1

ADCI	ADCS	Operation
0	0	ADC not busy, a conversion can be started.
0	1	ADC busy, start of a new conversion is blocked.
1	0	Conversion completed, start of a new conversion is blocked.
1	1	Not possible.

INPUT CHANNEL SELECTION			
ADDR2	ADDR1	ADDR0	INPUT PIN
0	0	0	P1.0
0	0	1	P1.1
0	1	0	P1.2
0	1	1	P1.3
1	0	0	P1.4

## Position Symbol

ADCON.5 ENADC

## Function

Enable A/D function when ENADC = 1. Reset forces ENADC = 0.

ADCON.4 ADCI

ADC interrupt flag. This flag is set when an ADC conversion is complete. If IE.6 = 1, an interrupt is requested when ADCI = 1. The ADCI flag is cleared when conversion data is read. This flag is read only.

ADCON.3 ADCS

ADC start. Setting this bit starts an A/D conversion. Once set, ADCS remains high throughout the conversion cycle. On completion of the conversion, it is reset just before the ADCI interrupt flag is cleared. ADCS cannot be reset by software. ADCS should not be used to monitor the A/D converter status. ADCI should be used for this purpose.

ADCON.2 AADR2

Analog input select.

ADCON.1 AADR1

Analog input select.

ADCON.0 AADR0

Analog input select.

This binary coded address selects one of the five analog input port pins of P1 to be input to the converter.

It can only be changed when ADCI and ADCS are both low. AADR2 is the most significant bit.

The completion of the 8-bit ADC conversion is flagged by ADCI in the ADCON register, and the result is stored in the special function register ADAT.

An ADC conversion in progress is unaffected by an ADC start. The result of a completed conversion remains unaffected provided ADCI remains at a logic 1. While ADCS is a logic 1 or ADCI is a logic 1, a new ADC START will be blocked and consequently lost. An ADC conversion in progress is aborted when the idle or power-down mode is entered. The result of a completed conversion (ADCI = logic 1) remains unaffected when entering the idle mode. See Figure 2 for an A/D input equivalent circuit.

The analog input pins ADC0-ADC4 may be used as digital inputs and outputs when the A/D converter is disabled by a 0 in the ENADC bit in ADCON. When the A/D is enabled, the analog input channel that is selected by the ADDR2-ADDR0 bits in ADCON cannot be used as a digital input. Reading the selected A/D channel as a digital input will always return a 1. The unselected A/D inputs may always be used as digital inputs. Unselected analog inputs will be floating and may not be used as digital outputs.

The A/D reference inputs on the 83C752 are tied together with the analog supply pins AV<sub>CC</sub> and AV<sub>SS</sub>. This means that the reference voltage on the A/D cannot be varied separately from the analog supply pins. AV<sub>SS</sub> must be connected to 0V and AV<sub>CC</sub> must be connected to a supply voltage between 4.5V and 5.5V. A/D measurements may be made in the range of 4.5V to 5.5V. Increasing the voltage on the A/D ground reference above 0V or reducing the voltage on the positive A/D reference below 4.5V is not permitted.

## CMOS single-chip 8-bit microcontroller 83C752/87C752

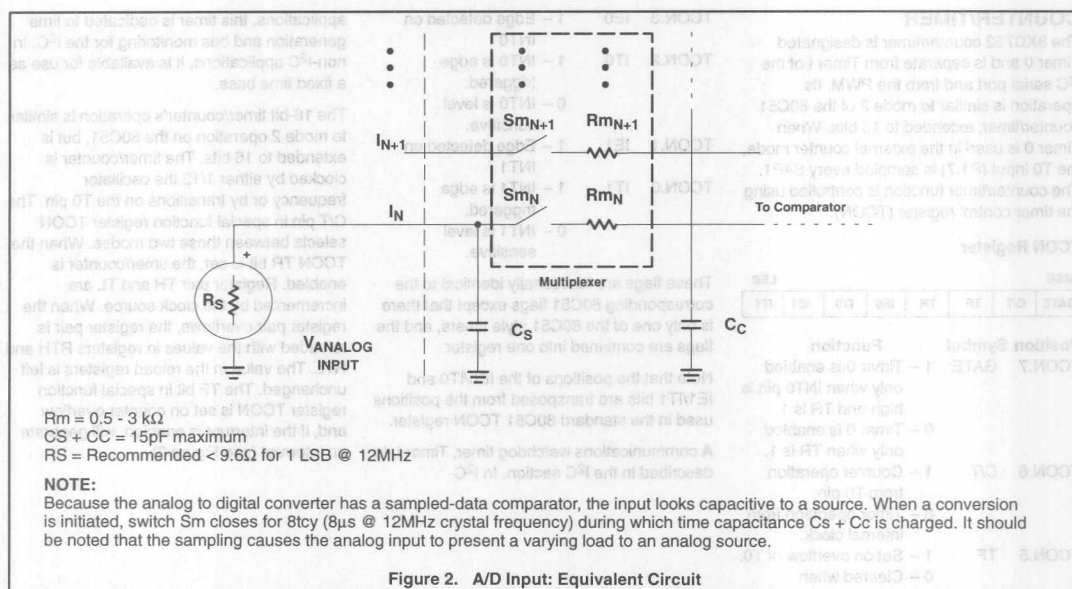


Figure 2. A/D Input: Equivalent Circuit

## A/D CONVERTER PARAMETER DEFINITIONS

The following definitions are included to clarify some specifications given and do not represent a complete set of A/D parameter definitions.

## Absolute Accuracy Error

Absolute accuracy error of a given output is the difference between the theoretical analog input voltage to produce a given output and the actual analog input voltage required to produce the same code. Since the same output code is produced by a band of input voltages, the "required input voltage" is defined as the midpoint of the band of input voltage that will produce that code. Absolute accuracy error not specified with a code is the maximum over all codes.

## Nonlinearity

If a straight line is drawn between the end points of the actual converter characteristics such that zero offset and full scale errors are removed, then non-linearity is the maximum deviation of the code transitions of the actual characteristics from that of the straight line so constructed. This is also referred to as relative accuracy and also integral non-linearity.

## Differential Non-Linearity

Differential non-linearity is the maximum difference between the actual and ideal code widths of the converter. The code widths are the differences expressed in LSB between the code transition points, as the input voltage is varied through the range for the complete set of codes.

## Gain Error

Gain error is the deviation between the ideal and actual analog input voltage required to cause the final code transition to a full-scale output code after the offset error has been removed. This may sometimes be referred to as full scale error.

## Offset Error

Offset error is the difference between the actual input voltage that causes the first code transition and the ideal value to cause the first code transition. This ideal value is 1/2 LSB above  $V_{ref-}$ .

## Channel to Channel Matching

Channel to channel matching is the maximum difference between the corresponding code transitions of the actual characteristics taken from different channels under the same

temperature, voltage and frequency conditions.

## Crosstalk

Crosstalk is the measured level of a signal at the output of the converter resulting from a signal applied to one deselected channel.

## Total Error

Maximum deviation of any step point from a line connecting the ideal first transition point to the ideal last transition point.

## Relative Accuracy

Relative accuracy error is the deviation of the ADC's actual code transition points from the ideal code transition points on a straight line which connects the ideal first code transition point and the final code transition point, after nullifying offset error and gain error. It is generally expressed in LSBs or in percent of FSR.

## CMOS single-chip 8-bit microcontroller 83C752/87C752

## COUNTER/TIMER

The 83C752 counter/timer is designated Timer 0 and is separate from Timer 1 of the I<sup>2</sup>C serial port and from the PWM. Its operation is similar to mode 2 of the 80C51 counter/timer, extended to 16 bits. When Timer 0 is used in the external counter mode, the T0 input (P1.7) is sampled every S4P1. The counter/timer function is controlled using the timer control register (TCON).

## TCON Register

MSB							LSB
GATE	C/T	TF	TR	IE0	IT0	IE1	IT1

Position	Symbol	Function
TCON.7	GATE	1 – Timer 0 is enabled only when INT0 pin is high and TR is 1. 0 – Timer 0 is enabled only when TR is 1.
TCON.6	C/T	1 – Counter operation from T0 pin. 0 – Timer operation from internal clock.
TCON.5	TF	1 – Set on overflow of T0. 0 – Cleared when processor vectors to interrupt routine and by reset.
TCON.4	TR	1 – Enable timer 0 0 – Disable timer 0

TCON.3	IE0	1 – Edge detected on INT0
TCON.2	IT0	1 – INT0 is edge triggered. 0 – INT0 is level sensitive.
TCON.1	IE1	1 – Edge detected on INT1
TCON.0	IT1	1 – INT1 is edge triggered. 0 – INT1 is level sensitive.

These flags are functionally identical to the corresponding 80C51 flags except that there is only one of the 80C51 style timers, and the flags are combined into one register.

Note that the positions of the IE0/IT0 and IE1/IT1 bits are transposed from the positions used in the standard 80C51 TCON register.

A communications watchdog timer, Timer 1, is described in the I<sup>2</sup>C section. In I<sup>2</sup>C

applications, this timer is dedicated to time generation and bus monitoring for the I<sup>2</sup>C. In non-I<sup>2</sup>C applications, it is available for use as a fixed time base.

The 16-bit timer/counter's operation is similar to mode 2 operation on the 80C51, but is extended to 16 bits. The timer/counter is clocked by either 1/12 the oscillator frequency or by transitions on the T0 pin. The C/T pin in special function register TCON selects between these two modes. When the TCON TR bit is set, the timer/counter is enabled. Register pair TH and TL are incremented by the clock source. When the register pair overflows, the register pair is reloaded with the values in registers RTH and RTL. The value in the reload registers is left unchanged. The TF bit in special function register TCON is set on counter overflow and, if the interrupt is enabled, will generate an interrupt (see Figure 3).

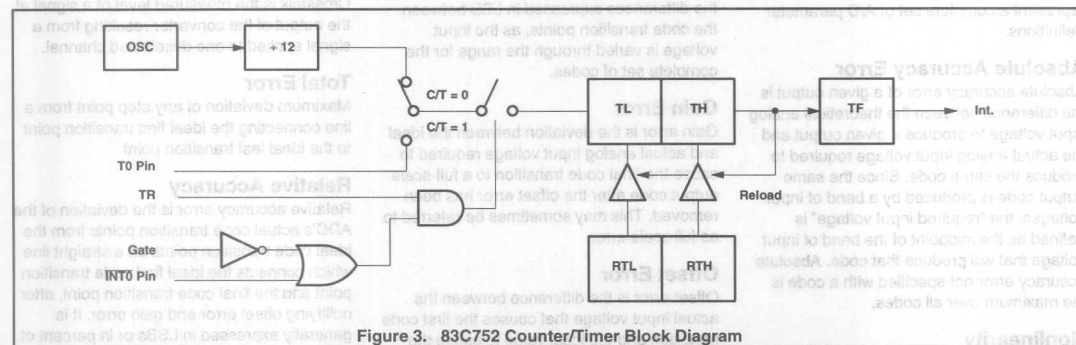


Figure 3. 83C752 Counter/Timer Block Diagram

Table 2. I<sup>2</sup>C Special Function Register Addresses

REGISTER ADDRESS				BIT ADDRESS					
NAME	SYMBOL	ADDRESS	MSB					LSB	
I <sup>2</sup> C control	I2CON	98	9F	9E	9D	9C	9B	9A	99
I <sup>2</sup> C data	I2DAT	99	–	–	–	–	–	–	–
I <sup>2</sup> C configuration	I2CFG	D8	DF	DE	DD	DC	DB	DA	D9
I <sup>2</sup> C status	I2STA	F8	FF	FE	FD	FC	FB	FA	F9



## CMOS single-chip 8-bit microcontroller 83C752/87C752

ABSOLUTE MAXIMUM RATINGS<sup>1, 3, 4</sup>

PARAMETER	RATING	UNIT
Storage temperature range	-65 to +150	°C
Voltage from $V_{CC}$ to $V_{SS}$	-0.5 to +6.5	V
Voltage from any pin to $V_{SS}$ (except $V_{PP}$ )	-0.5 to $V_{CC} + 0.5$	V
Power dissipation	1.0	W
Voltage from $V_{PP}$ pin to $V_{SS}$	-0.5 to +13.0	V

## DC ELECTRICAL CHARACTERISTICS

$T_{amb} = 0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$  or  $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $AV_{CC} = 5\text{V} \pm 5\%$ ,  $AV_{SS} = 0\text{V}^4$

83C752:  $V_{CC} = 5\text{V} \pm 20\%$ , 87C752:  $V_{CC} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS <sup>4</sup>			UNIT
			MIN	Typical <sup>1</sup>	MAX	
$I_{CC}$	Supply current (see Figure 6)					mA
<b>Inputs</b>						
$V_{IL}$	Input low voltage, except SDA, SCL	(0 to $70^{\circ}\text{C}$ ) (-40 to $+85^{\circ}\text{C}$ )	-0.5 -0.5		$0.2V_{CC}-0.1$ $0.2V_{CC}-0.15$	V
$V_{IH}$	Input high voltage, except X1, RST	(0 to $70^{\circ}\text{C}$ ) ( $0.2V_{CC}+1$ )	$0.2V_{CC}+0.9$ (-40 to $+85^{\circ}\text{C}$ )		$V_{CC}+0.5$ $V_{CC}+0.5$	V
$V_{IH1}$	Input high voltage, X1, RST	(0 to $70^{\circ}\text{C}$ ) (-40 to $+85^{\circ}\text{C}$ )	$0.7V_{CC}$ $0.7V_{CC}+0.1$		$V_{CC}+0.5$ $V_{CC}+0.5$	V
$V_{IL1}$	SDA, SCL, P0.2 Input low voltage	(0 to $70^{\circ}\text{C}$ ) (-40 to $+85^{\circ}\text{C}$ )	-0.5 -0.5		$0.3V_{CC}$ $0.3V_{CC}-0.1$	V
$V_{IH2}$	Input high voltage	(0 to $70^{\circ}\text{C}$ ) (-40 to $+85^{\circ}\text{C}$ )	$0.7V_{CC}$ $0.7V_{CC}+0.1$		$V_{CC}+0.5$ $V_{CC}+0.5$	V
<b>Outputs</b>						
$V_{OL}$	Output low voltage, ports 1, 3, 0.3, and 0.4 (PWM disabled)	$I_{OL} = 1.6\text{mA}^2$			0.45	V
$V_{OL1}$	Output low voltage, port 0.2	$I_{OL} = 3.2\text{mA}^2$			0.45	V
$V_{OH}$	Output high voltage, ports 1, 3, 0.3, and 0.4 (PWM disabled)	$I_{OH} = -60\mu\text{A}$ $I_{OH} = -25\mu\text{A}$ $I_{OH} = -10\mu\text{A}$	2.4 $0.75V_{CC}$ $0.9V_{CC}$		V V V	
$V_{OH2}$	Output high voltage, P0.4 (PWM enabled)	$I_{OH} = -400\mu\text{A}$ $I_{OH} = -40\mu\text{A}$	2.4 $0.9V_{CC}$		V V	
$V_{OL2}$	Port 0.0 and 0.1 ( $I^2\text{C}$ ) – Drivers Output low voltage	$I_{OL} = 3\text{mA}$ (over $V_{CC}$ range)			0.4	V
C	Driver, receiver combined: Capacitance				10	pF
$I_{IL}$	Logical 0 input current, ports 1, 3, 0.3, and 0.4 (PWM disabled) <sup>11</sup>	$V_{IN} = 0.45\text{V}$ (0 to $70^{\circ}\text{C}$ ) $V_{IN} = 0.45\text{V}$ (0 to $+85^{\circ}\text{C}$ )			-50 -75	$\mu\text{A}$ $\mu\text{A}$
$I_{TL}$	Logical 1 to 0 transition current, ports 1, 3, 0.3 and 0.4 <sup>11</sup>	$V_{IN} = 2\text{V}$ (0 to $70^{\circ}\text{C}$ ) $V_{IN} = 2\text{V}$ (-40 to $+85^{\circ}\text{C}$ )			-650 -750	$\mu\text{A}$ $\mu\text{A}$
$I_{LI}$	Input leakage current, port 0.0, 0.1 and 0.2	$0.45 < V_{IN} < V_{CC}$			$\pm 10$	$\mu\text{A}$
$R_{RST}$	Reset pull-down resistor		25		175	k $\Omega$
$C_{IO}$	Pin capacitance	Test freq = 1MHz, $T_{amb} = 25^{\circ}\text{C}$			10	pF
$I_{PD}$	Power-down current <sup>5</sup>	$V_{CC} = 2$ to $5.5\text{V}$ $V_{CC} = 2$ to $6.0\text{V}$ (83C752)			50	$\mu\text{A}$
$V_{PP}$	$V_{PP}$ program voltage (87C752 only)	$V_{SS} = 0\text{V}$ $V_{CC} = 5\text{V} \pm 10\%$ $T_{amb} = 21^{\circ}\text{C}$ to $27^{\circ}\text{C}$	12.5		13.0	V
$I_{PP}$	Program current (87C752 only)	$V_{PP} = 13.0\text{V}$			50	mA

## CMOS single-chip 8-bit microcontroller 83C752/87C752

## DC ELECTRICAL CHARACTERISTICS (Continued)

$T_{amb} = 0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$  or  $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $AV_{CC} = 5\text{V} \pm 5\%$ ,  $AV_{SS} = 0\text{V}^4$

83C752:  $V_{CC} = 5\text{V} \pm 20\%$ , 87C752:  $V_{CC} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS <sup>4</sup>			UNIT
			MIN	Typical <sup>1</sup>	MAX	
Analog Inputs (A/D guaranteed only with quartz window covered.)						
AV <sub>CC</sub>	Analog supply voltage <sup>10</sup>	AV <sub>CC</sub> = V <sub>CC</sub> ±0.2V	4.5		5.5	V
AI <sub>CC</sub>	Analog operating supply current	AV <sub>CC</sub> = 5.12V			3 <sup>9</sup>	mA
AV <sub>IN</sub>	Analog input voltage		AV <sub>SS</sub> -0.2		AV <sub>CC</sub> +0.2	V
C <sub>IA</sub>	Analog input capacitance				15	pF
t <sub>ADS</sub>	Sampling time				8t <sub>CY</sub>	s
t <sub>ADC</sub>	Conversion time				40t <sub>CY</sub>	s
R	Resolution				8	bits
E <sub>RA</sub>	Relative accuracy				±1	LSB <sup>5</sup>
OS <sub>e</sub>	Zero scale offset				±1	LSB
G <sub>e</sub>	Full scale gain error				0.4	%
M <sub>CTC</sub>	Channel to channel matching				±1	LSB
C <sub>t</sub>	Crosstalk	0–100kHz			–60	dB

## NOTES:

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
- Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:  
 Maximum  $I_{OL}$  per port pin: 10mA (NOTE: This is 85°C spec.)  
 Maximum  $I_{OL}$  per 8-bit port: 26mA  
 Maximum total  $I_{OL}$  for all outputs: 67mA  
 If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
- This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
- Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to  $V_{SS}$  unless otherwise noted.
- Power-down  $I_{CC}$  is measured with all output pins disconnected; port 0 =  $V_{CC}$ ; X2, X1 n.c.; RST =  $V_{SS}$ .
- $I_{CC}$  is measured with all output pins disconnected; X1 driven with  $t_{CLCH}$ ,  $t_{CHCL} = 5\text{ns}$ ,  $V_{IL} = V_{SS} + 0.5\text{V}$ ,  $V_{IH} = V_{CC} - 0.5\text{V}$ ; X2 n.c.; RST = port 0 =  $V_{CC}$ .  $I_{CC}$  will be slightly higher if a crystal oscillator is used.
- Idle  $I_{CC}$  is measured with all output pins disconnected; X1 driven with  $t_{CLCH}$ ,  $t_{CHCL} = 5\text{ns}$ ,  $V_{IL} = V_{SS} + 0.5\text{V}$ ,  $V_{IH} = V_{CC} - 0.5\text{V}$ ; X2 n.c.; port 0 =  $V_{CC}$ ; RST =  $V_{SS}$ .
- Load capacitance for ports = 80pF.
- The resistor ladder network is not disconnected in the power down or idle modes. Thus, to conserve power, the user may remove  $AV_{CC}$ .
- If the A/D function is not required, or if the A/D function is only needed periodically,  $AV_{CC}$  may be removed without affecting the operation of the digital circuitry. Contents of ADCON and ADAT are not guaranteed to be valid. If  $AV_{CC}$  is removed, the A/D inputs must be lowered to less than 0.5V. Digital inputs on P1.0–P1.4 will not function normally.
- These parameters do not apply to P1.0–P1.4 if the A/D function is enabled.

## CMOS single-chip 8-bit microcontroller 83C752/87C752

## AC ELECTRICAL CHARACTERISTICS

$T_{amb} = 0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$  or  $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$  (87C752),  $V_{CC} = 5\text{V} \pm 20\%$  (83C752),  $V_{SS} = 0\text{V}^{4,8}$

SYMBOL	PARAMETER	12MHz CLOCK		VARIABLE CLOCK		UNIT
		MIN	MAX	MIN	MAX	
$1/t_{CLCL}$	Oscillator frequency:			3.5	12	MHz
				3.5	16	MHz
External Clock (Figure 4)						
$t_{CHCX}$	High time	20		20		ns
$t_{CLCX}$	Low time	20		20		ns
$t_{CLCH}$	Rise time		20		20	ns
$t_{CHCL}$	Fall time		20		20	ns

## EXPLANATION OF THE AC SYMBOLS

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

C – Clock  
D – Input data

H – Logic level high  
L – Logic level low  
Q – Output data  
T – Time  
V – Valid  
X – No longer a valid logic level  
Z – Float

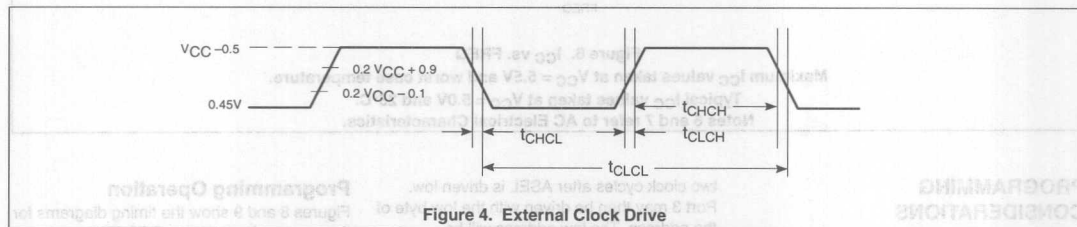


Figure 4. External Clock Drive

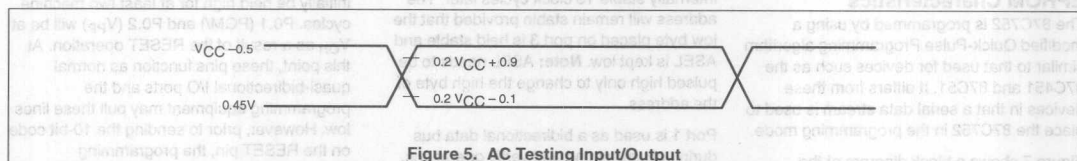
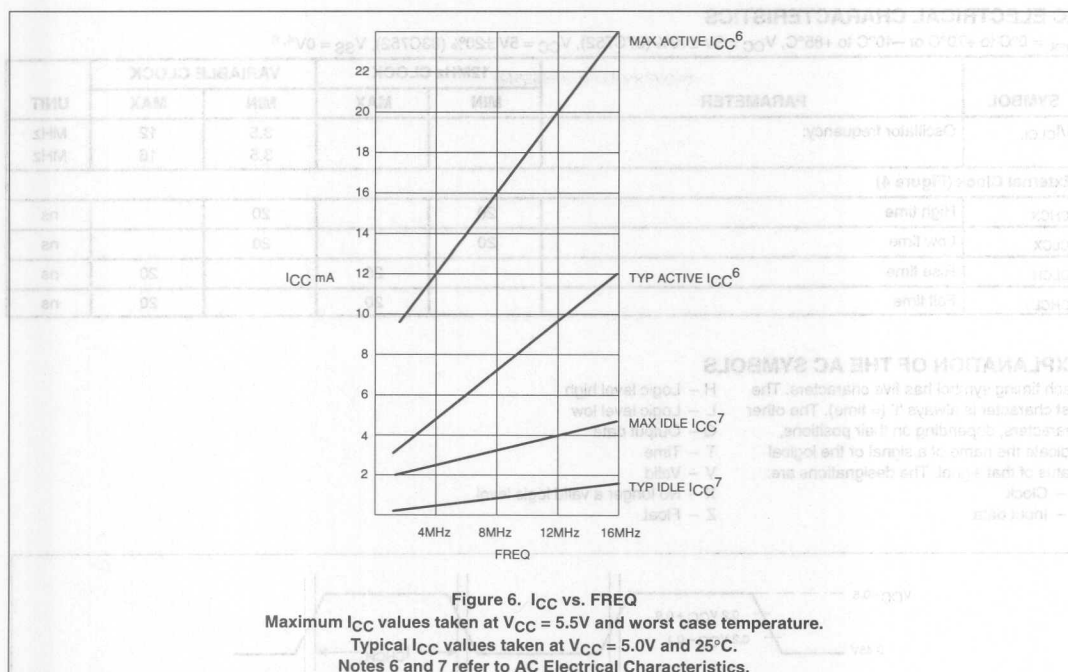


Figure 5. AC Testing Input/Output

## CMOS single-chip 8-bit microcontroller 83C752/87C752



## PROGRAMMING CONSIDERATIONS

## EPROM Characteristics

The 87C752 is programmed by using a modified Quick-Pulse Programming algorithm similar to that used for devices such as the 87C451 and 87C51. It differs from these devices in that a serial data stream is used to place the 87C752 in the programming mode.

Figure 7 shows a block diagram of the programming configuration for the 87C752. Port pin P0.2 is used as the programming voltage supply input ( $V_{PP}$  signal). Port pin P0.1 is used as the program (PGM) signal. This pin is used for the 25 programming pulses.

Port 3 is used as the address input for the byte to be programmed and accepts both the high and low components of the eleven bit address. Multiplexing of these address components is performed using the ASEL input. The user should drive the ASEL input high and then drive port 3 with the high order bits of the address. ASEL should remain high for at least 13 clock cycles. ASEL may then be driven low which latches the high order bits of the address internally. The high address should remain on port 3 for at least

two clock cycles after ASEL is driven low. Port 3 may then be driven with the low byte of the address. The low address will be internally stable 13 clock cycles later. The address will remain stable provided that the low byte placed on port 3 is held stable and ASEL is kept low. **Note:** ASEL needs to be pulsed high only to change the high byte of the address.

Port 1 is used as a bidirectional data bus during programming and verify operations. During programming mode, it accepts the byte to be programmed. During verify mode, it provides the contents of the EPROM location specified by the address which has been supplied to Port 3.

The XTAL1 pin is the oscillator input and receives the master system clock. This clock should be between 1.2 and 6MHz.

The RESET pin is used to accept the serial data stream that places the 87C752 into various programming modes. This pattern consists of a 10-bit code with the LSB sent first. Each bit is synchronized to the clock input, X1.

## Programming Operation

Figures 8 and 9 show the timing diagrams for the program/verify cycle. RESET should initially be held high for at least two machine cycles. P0.1 (PGM) and P0.2 ( $V_{PP}$ ) will be at  $V_{OH}$  as a result of the RESET operation. At this point, these pins function as normal quasi-bidirectional I/O ports and the programming equipment may pull these lines low. However, prior to sending the 10-bit code on the RESET pin, the programming equipment should drive these pins high ( $V_{IH}$ ). The RESET pin may now be used as the serial data input for the data stream which places the 87C752 in the programming mode. Data bits are sampled during the clock high time and thus should only change during the time that the clock is low. Following transmission of the last data bit, the RESET pin should be held low.

Next the address information for the location to be programmed is placed on port 3 and ASEL is used to perform the address multiplexing, as previously described. At this time, port 1 functions as an output.

## CMOS single-chip 8-bit microcontroller 83C752/87C752

A high voltage  $V_{PP}$  level is then applied to the  $V_{PP}$  input (P0.2). (This sets Port 1 as an input port). The data to be programmed into the EPROM array is then placed on Port 1. This is followed by a series of programming pulses applied to the PGM/ pin (P0.1). These pulses are created by driving P0.1 low and then high. This pulse is repeated until a total of 25 programming pulses have occurred. At the conclusion of the last pulse, the PGM/ signal should remain high.

The  $V_{PP}$  signal may now be driven to the  $V_{OH}$  level, placing the 87C752 in the verify mode. (Port 1 is now used as an output port). After four machine cycles (48 clock periods), the contents of the addressed location in the EPROM array will appear on Port 1.

The next programming cycle may now be initiated by placing the address information at the inputs of the multiplexed buffers, driving the  $V_{PP}$  pin to the  $V_{PP}$  voltage level, providing the byte to be programmed to Port 1 and issuing the 26 programming pulses on the PGM/ pin, bringing  $V_{PP}$  back down to the  $V_C$  level and verifying the byte.

### Programming Modes

The 87C752 has four programming features incorporated within its EPROM array. These include the USER EPROM for storage of the application's code, a 16-byte encryption key array and two security bits. Programming and verification of these four elements are selected by a combination of the serial data stream applied to the RESET pin and the voltage levels applied to port pins P0.1 and P0.2. The various combinations are shown in Table 3.

### Encryption Key Table

The 87C752 includes a 16-byte EPROM array that is programmable by the end user. The contents of this array can then be used to encrypt the program memory contents during a program memory verify operation. When a program memory verify operation is performed, the contents of the program

memory location is XNOR'ed with one of the bytes in the 16-byte encryption table. The resulting data pattern is then provided to port 1 as the verify data. The encryption mechanism can be disabled, in essence, by leaving the bytes in the encryption table in their erased state (FFH) since the XNOR product of a bit with a logical one will result in the original bit. The encryption bytes are mapped with the code memory in 16-byte groups: the first byte in code memory will be encrypted with the first byte in the encryption table; the second byte in code memory will be encrypted with the second byte in the encryption table and so forth up to and including the 16th byte. The encryption repeats in 16-byte groups; the 17th byte in the code memory will be encrypted with the first byte in the encryption table, and so forth.

### Security Bits

Two security bits, security bit 1 and security bit 2, are provided to limit access to the USER EPROM and encryption key arrays. Security bit 1 is the program inhibit bit, and once programmed performs the following functions:

1. Additional programming of the USER EPROM is inhibited.
2. Additional programming of the encryption key is inhibited.
3. Verification of the encryption key is inhibited.
4. Verification of the USER EPROM and the security bit levels may still be performed.

(If the encryption key array is being used, this security bit should be programmed by the user to prevent unauthorized parties from reprogramming the encryption key to all logical zero bits. Such programming would provide data during a verify cycle that is the logical complement of the USER EPROM contents).

Security bit 2, the verify inhibit bit, prevents verification of both the USER EPROM array

and the encryption key arrays. The security bit levels may still be verified.

### Programming and Verifying Security Bits

Security bits are programmed employing the same techniques used to program the USER EPROM and KEY arrays using serial data streams and logic levels on port pins indicated in Table 3. When programming either security bit, it is not necessary to provide address or data information to the 87C752 on ports 1 and 3.

Verification occurs in a similar manner using the RESET serial stream shown in Table 3. Port 3 is not required to be driven and the results of the verify operation will appear on ports 1.6 and 1.7.

Ports 1.7 contains the security bit 1 data and is a logical one if programmed and a logical zero if erased. Likewise, P1.6 contains the security bit 2 data and is a logical one if programmed and a logical zero if erased.

### Erase Characteristics

Erase of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent erasure. **For this and secondary effects, it is recommended that an opaque label be placed over the window.** For elevated temperature or environments where solvents are being used, apply Kapton tape Flourless part number 2345-5 or equivalent.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 angstroms) to an integrated dose of at least 15W-sec/cm<sup>2</sup>. Exposing the EPROM to an ultraviolet lamp of 12,000uW/cm<sup>2</sup> rating for 20 to 39 minutes, at a distance of about 1 inch, should be sufficient.

Erase leaves the array in an all 1s state.

Table 3. Implementing Program/Verify Modes

OPERATION	SERIAL CODE	P0.1 (PGM/)	P0.2 ( $V_{PP}$ )
Program user EPROM	296H	—*	$V_{PP}$
Verify user EPROM	296H	$V_{IH}$	$V_{IH}$
Program key EPROM	292H	—*	$V_{PP}$
Verify key EPROM	292H	$V_{IH}$	$V_{IH}$
Program security bit 1	29AH	—*	$V_{PP}$
Program security bit 2	298H	—*	$V_{PP}$
Verify security bits	29AH	$V_{IH}$	$V_{IH}$

#### NOTE:

\* Pulsed from  $V_{IH}$  to  $V_{IL}$  and returned to  $V_{IH}$ .



CMOS single-chip 8-bit microcontroller 83C752/87C752

EPROM PROGRAMMING AND VERIFICATION

T<sub>amb</sub> = 21°C to +27°C, V<sub>CC</sub> = 5V ±10%, V<sub>SS</sub> = 0V

SYMBOL	PARAMETER	MIN	MAX	UNIT
1/t <sub>CLCL</sub>	Oscillator/clock frequency	1.2	6	MHz
t <sub>AVGL</sub> <sup>1</sup>	Address setup to P0.1 (PROG-) low	10μs + 24t <sub>CLCL</sub>		
t <sub>GHAX</sub>	Address hold after P0.1 (PROG-) high	48t <sub>CLCL</sub>		
t <sub>DVGL</sub>	Data setup to P0.1 (PROG-) low	38t <sub>CLCL</sub>		
t <sub>DVGL</sub>	Data setup to P0.1 (PROG-) low	38t <sub>CLCL</sub>		
t <sub>GHDX</sub>	Data hold after P0.1 (PROG-) high	36t <sub>CLCL</sub>		
t <sub>SHGL</sub>	V <sub>PP</sub> setup to P0.1 (PROG-) low	10		μs
t <sub>GHSL</sub>	V <sub>PP</sub> hold after P0.1 (PROG-)	10		μs
t <sub>GLGH</sub>	P0.1 (PROG-) width	90	110	μs
t <sub>AVQV</sub> <sup>2</sup>	V <sub>PP</sub> low (V <sub>CC</sub> ) to data valid		48t <sub>CLCL</sub>	
t <sub>GHGL</sub>	P0.1 (PROG-) high to P0.1 (PROG-) low	10		μs
t <sub>SYNL</sub>	P0.0 (sync pulse) low	4t <sub>CLCL</sub>		
t <sub>SYNH</sub>	P0.0 (sync pulse) high	8t <sub>CLCL</sub>		
t <sub>MASEL</sub>	ASEL high time	13t <sub>CLCL</sub>		
t <sub>MAHLD</sub>	Address hold time	2t <sub>CLCL</sub>		
t <sub>HASET</sub>	Address setup to ASEL	13t <sub>CLCL</sub>		
t <sub>ADSTA</sub>	Low address to address stable	13t <sub>CLCL</sub>		

NOTES:

- Address should be valid at least 24t<sub>CLCL</sub> before the rising edge of P0.2 (V<sub>PP</sub>).
- For a pure verify mode, i.e., no program mode in between, t<sub>AVQV</sub> is 14t<sub>CLCL</sub> maximum.

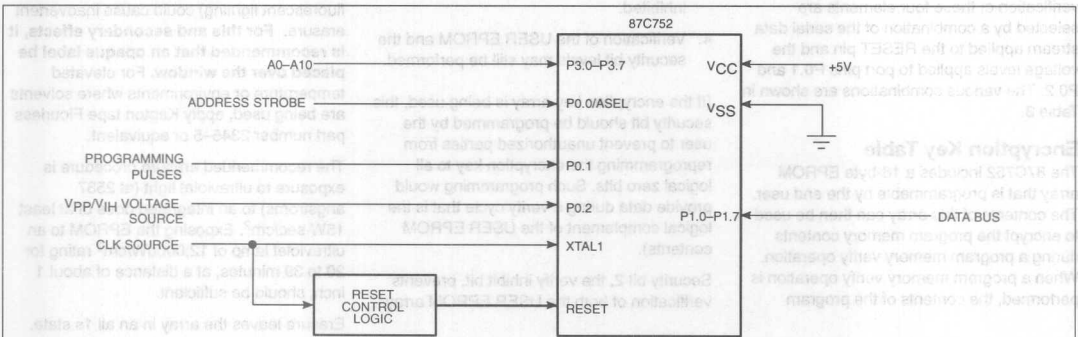
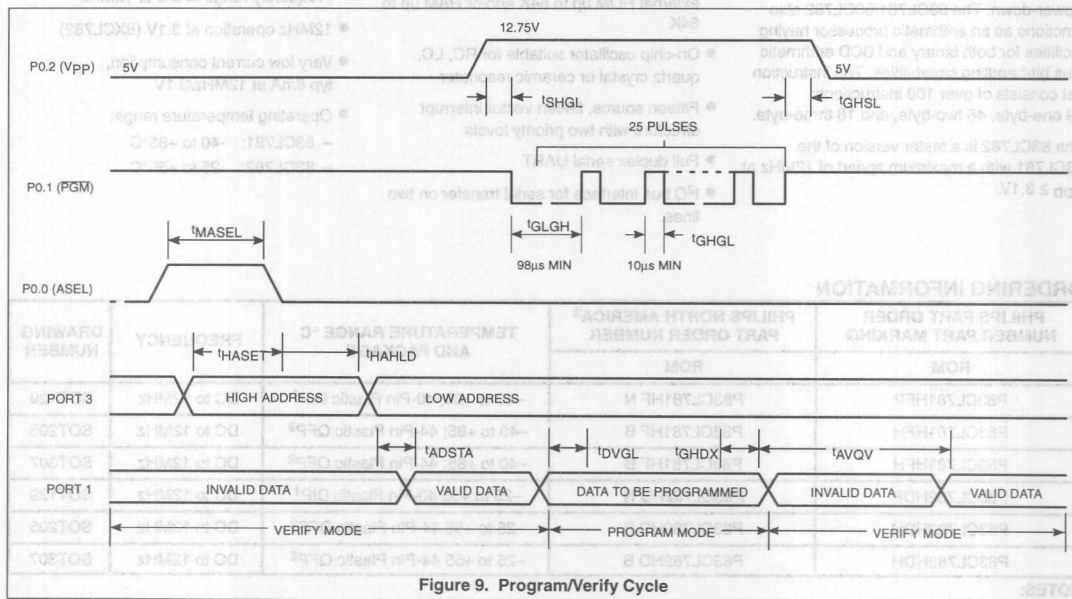
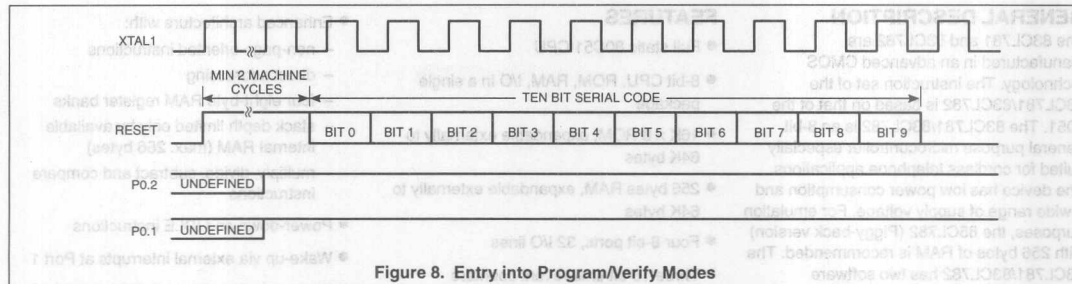


Figure 7. Programming Configuration

OPERATION	SERIAL CODE	P0.1 (PROG-)	P0.2 (V <sub>PP</sub> )
Program user EPROM	328H	V <sub>SS</sub>	V <sub>PP</sub>
Verify user EPROM	329H	V <sub>SS</sub>	V <sub>PP</sub>
Program key EPROM	32AH	V <sub>SS</sub>	V <sub>PP</sub>
Verify key EPROM	32BH	V <sub>SS</sub>	V <sub>PP</sub>
Program security bit 1	32CH	V <sub>SS</sub>	V <sub>PP</sub>
Program security bit 2	32DH	V <sub>SS</sub>	V <sub>PP</sub>
Verify security bit	32EH	V <sub>SS</sub>	V <sub>PP</sub>

## CMOS single-chip 8-bit microcontroller

83C752/87C752



Purchase of Philips' I<sup>2</sup>C components conveys a license under the Philips' I<sup>2</sup>C patent to use the components in the I<sup>2</sup>C-system provided the system conforms to the I<sup>2</sup>C specifications defined by Philips.

## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

### GENERAL DESCRIPTION

The 83CL781 and 83CL782 are manufactured in an advanced CMOS technology. The instruction set of the 83CL781/83CL782 is based on that of the 8051. The 83CL781/83CL782 is an 8-bit general purpose microcontroller especially suited for cordless telephone applications. The device has low power consumption and a wide range of supply voltage. For emulation purposes, the 85CL782 (Piggy-back version) with 256 bytes of RAM is recommended. The 83CL781/83CL782 has two software selectable modes of reduced activity for further power reduction: Idle and Power-down. The 83CL781/83CL782 also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of over 100 instructions: 49 one-byte, 46 two-byte, and 16 three-byte.

The 83CL782 is a faster version of the 83CL781 with a maximum speed of 12MHz at  $V_{DD} \geq 3.1V$ .

### FEATURES

- Full static 80C51 CPU
- 8-bit CPU, ROM, RAM, I/O in a single package
- 16K  $\times$  8 ROM, expandable externally to 64K bytes
- 256 bytes RAM, expandable externally to 64K bytes
- Four 8-bit ports, 32 I/O lines
- Three 16-bit timer/event counters
- External memory expandable up to 128K, external ROM up to 64K and/or RAM up to 64K
- On-chip oscillator suitable for RC, LC, quartz crystal or ceramic resonator
- Fifteen source, fifteen vector interrupt structure with two priority levels
- Full duplex serial UART
- I<sup>2</sup>C bus interface for serial transfer on two lines.

- Enhanced architecture with:
  - non-page oriented instructions
  - direct addressing
  - four eight-byte RAM register banks
  - stack depth limited only by available internal RAM (max. 256 bytes)
  - multiply, divide, subtract and compare instructions
- Power-down and IDLE instructions
- Wake-up via external interrupts at Port 1
- Single supply voltage of 1.8V to 6.0V
- Frequency range of DC to 12MHz
- 12MHz operation at 3.1V (83CL782)
- Very low current consumption, typ 8mA at 12MHz/3.1V
- Operating temperature range:
  - 83CL781: -40 to +85°C
  - 83CL782: -25 to +55°C

### ORDERING INFORMATION

PHILIPS PART ORDER NUMBER PART MARKING	PHILIPS NORTH AMERICA <sup>3</sup> PART ORDER NUMBER	TEMPERATURE RANGE °C AND PACKAGE	FREQUENCY	DRAWING NUMBER
ROM	ROM			
P83CL781HFP	P83CL781HF N	-40 to +85; 40-Pin Plastic DIP <sup>1</sup>	DC to 12MHz	SOT129
P83CL781HFH	P83CL781HF B	-40 to +85; 44-Pin Plastic QFP <sup>2</sup>	DC to 12MHz	SOT205
P83CL781HFH	P83CL781HF B	-40 to +85; 44-Pin Plastic QFP <sup>2</sup>	DC to 12MHz	SOT307
P83CL782HDP	P83CL782HD N	-25 to +55 40-Pin Plastic DIP <sup>1</sup>	DC to 12MHz	SOT129
P83CL782HDH	P83CL782HD B	-25 to +55 44-Pin Plastic QFP <sup>2</sup>	DC to 12MHz	SOT205
P83CL782HDH	P83CL782HD B	-25 to +55 44-Pin Plastic QFP <sup>2</sup>	DC to 12MHz	SOT307

#### NOTES:

1. DIP = Dual In-line Package
2. QFP = Quad Flat Pack.

Two body sizes are available: SOT205 – 14mm  $\times$  14mm and SOT307 – 10mm  $\times$  10mm. See package dimension section for details.

3. Parts ordered by the Philips North America part number will be marked with the Philips part marking.

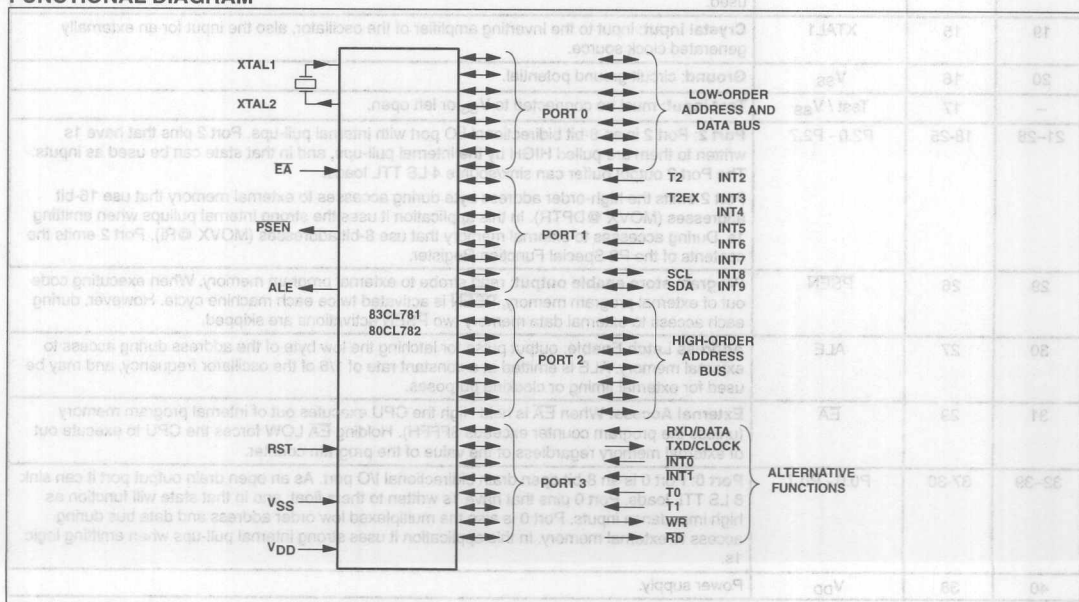


## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

## PIN CONFIGURATIONS

PIN DESCRIPTION		FUNCTION	
PIN NUMBER	DESCRIPTION	FUNCTION	PIN NUMBER
1	INT2/T2/P1.0	VDD	40
2	INT3/T2EX/P1.1	P0.0/AD0	39
3	INT4/P1.2	P0.1/AD1	38
4	INT5/P1.3	P0.2/AD2	37
5	INT6/P1.4	P0.3/AD3	36
6	INT7/P1.5	P0.4/AD4	35
7	SCL/INT8/P1.6	P0.5/AD5	34
8	SDA/INT9/P1.7	P0.6/AD6	33
9	RST	P0.7/AD7	32
10	RxD/data/P3.0	EA	31
11	TxD/clock/P3.1	ALE	30
12	INT0/P3.2	PSEN	29
13	INT1/P3.3	P2.7/A15	28
14	T0/P3.4	P2.6/A14	27
15	T1/P3.5	P2.5/A13	26
16	WR/P3.6	P2.4/A12	25
17	RD/P3.7	P2.3/A11	24
18	XTAL2	P2.2/A10	23
19	XTAL1	P2.1/A9	22
20	VSS	P2.0/A8	21
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
31			
32			
33			
34			
35			
36			
37			
38			
39			
40			

## FUNCTIONAL DIAGRAM



# Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

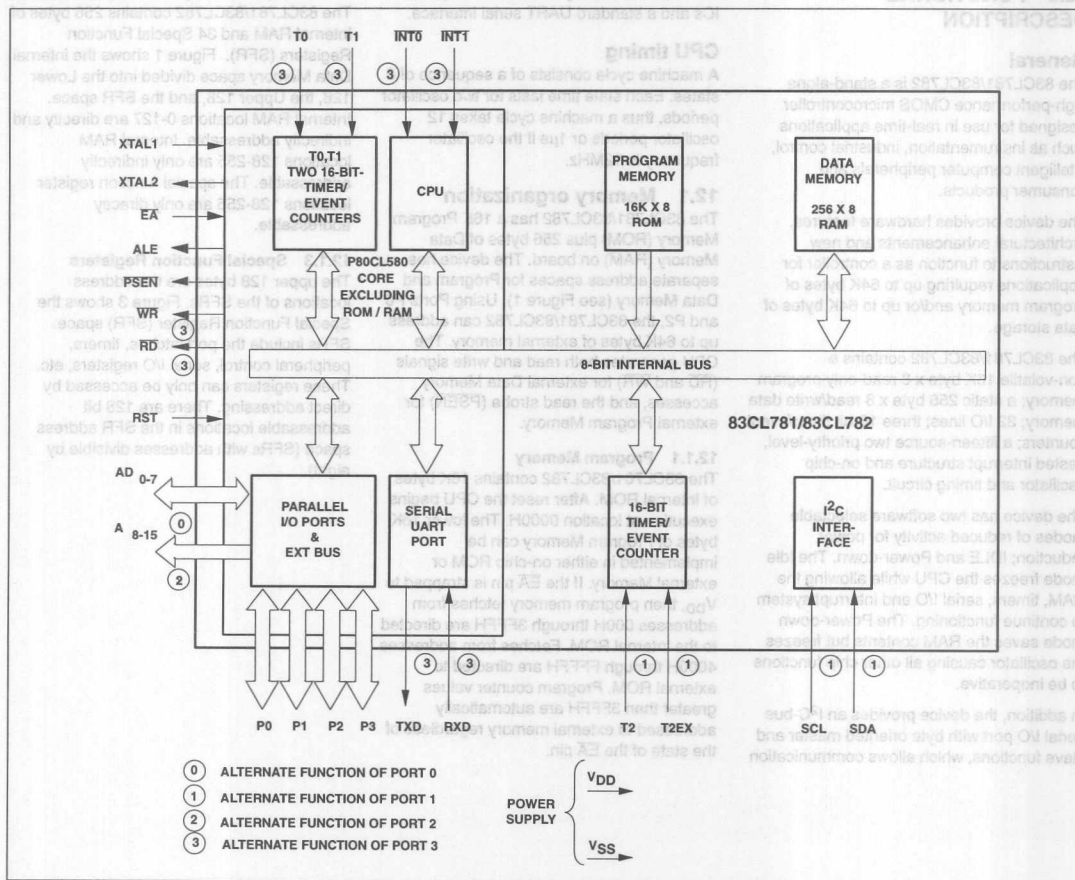
## PIN DESCRIPTION

PIN NUMBER		DESIGNATION	FUNCTION
DIP	QFP		
1	40	P1.0/INT2/T2	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pullups. Port 1 pins that have 1s written to them are pulled HIGH by the internal pullups, and in that state can be used as inputs. The Port 1 output buffer can sink/source 4 LS TTL loads. As inputs, Port 1 pins that are externally pulled LOW will source current ( $I_{IL}$ in the characteristics) due to the internal pullups. Port 1 also serves the alternative functions INT2 to INT9, and Timer T2 external input.
2	41	P1.1/INT3/T2EX	
3	42	P1.2/INT4	
4	43	P1.3/INT5	
5	44	P1.4/INT6	
6	1	P1.5/INT7	
7	2	P1.6/INT8	
8	3	P1.7/INT9	
9	4	RST	<b>Reset:</b> A high level on this pin for two machine cycles while the oscillator is running resets the device.
10–17	5, 7–13	P3.0 - P3.7	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 3 output buffers can sink/source 4 LS TTL inputs. Port 3 pins that have 1s written to them are pulled HIGH by the internal pull-ups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally pulled LOW will source current ( $I_{IL}$ in the characteristics) due to the internal pull-ups.
10	5	P3.0/RxD/data	<b>RxD/data:</b> serial port receiver data input (asynchronous) or data input/output (synchronous)
11	7	P3.1/TxD/clock	<b>TxD/clock:</b> serial port transmitter data output (asynchronous) or clock output (synchronous)
12	8	P3.2/INT0	<b>INT0:</b> external interrupt 0.
13	9	P3.3/INT1	<b>INT1:</b> external interrupt 1.
14	10	P3.4/T0	<b>T0:</b> Timer 0 external input.
15	11	P3.5/T1	<b>T1:</b> Timer 1 external input.
16	12	P3.6/WR	<b>WR:</b> external data memory write strobe.
17	13	P3.7/RD	<b>RD:</b> external data memory read strobe.
18	14	XTAL2	<b>Crystal output:</b> output of the inverting amplifier of the oscillator. Left open when external clock is used.
19	15	XTAL1	<b>Crystal input:</b> input to the inverting amplifier of the oscillator, also the input for an externally generated clock source.
20	16	V <sub>SS</sub>	<b>Ground:</b> circuit ground potential.
–	17	Test / V <sub>SS</sub>	<b>Test input:</b> must be connected to V <sub>SS</sub> or left open.
21–28	18–25	P2.0 - P2.7	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled HIGH by the internal pull-ups, and in that state can be used as inputs. The Port 2 output buffer can sink/source 4 LS TTL loads. Port 2 emits the high-order address byte during accesses to external memory that use 16-bit addresses (MOVX @DPTR). In this application it uses the strong internal pullups when emitting 1s. During accesses to external memory that use 8-bit addresses (MOVX @Ri). Port 2 emits the contents of the P2 Special Function Register.
29	26	PSEN	<b>Program store enable output:</b> read strobe to external program memory. When executing code out of external program memory, PSEN is activated twice each machine cycle. However, during each access to external data memory two PSEN activations are skipped.
30	27	ALE	<b>Address Latch Enable:</b> output pulse for latching the low byte of the address during access to external memory. ALE is emitted at a constant rate of 1/6 of the oscillator frequency, and may be used for external timing or clocking purposes.
31	29	EA	<b>External Access:</b> When EA is held High the CPU executes out of internal program memory (unless the program counter exceeds 3FFFH). Holding EA LOW forces the CPU to execute out of external memory regardless of the value of the program counter.
32–39	37–30	P0.0 - P0.7	<b>Port 0:</b> Port 0 is an 8-bit open drain bidirectional I/O port. As an open drain output port it can sink 8 LS TTL loads. Port 0 pins that have 1s written to them float, and in that state will function as high impedance inputs. Port 0 is also the multiplexed low order address and data bus during access to external memory. In this application it uses strong internal pull-ups when emitting logic 1s.
40	38	V <sub>DD</sub>	<b>Power supply.</b>



## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

## BLOCK DIAGRAM



## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

### 12.0 FUNCTIONAL DESCRIPTION

#### General

The 83CL781/83CL782 is a stand-alone high-performance CMOS microcontroller designed for use in real-time applications such as instrumentation, industrial control, intelligent computer peripherals and consumer products.

The device provides hardware features, architectural enhancements and new instructions to function as a controller for applications requiring up to 64K bytes of program memory and/or up to 64K bytes of data storage.

The 83CL781/83CL782 contains a non-volatile 16K byte x 8 read-only program memory; a static 256 byte x 8 read/write data memory; 32 I/O lines; three 16-bit timer/event counters; a fifteen-source two priority-level, nested interrupt structure and on-chip oscillator and timing circuit.

The device has two software selectable modes of reduced activity for power reduction; IDLE and Power-down. The Idle mode freezes the CPU while allowing the RAM, timers, serial I/O and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator causing all other chip functions to be inoperative.

In addition, the device provides an I<sup>2</sup>C-bus serial I/O port with byte oriented master and slave functions, which allows communication

with the whole family of I<sup>2</sup>C-bus compatible ICs and a standard UART serial interface.

#### CPU timing

A machine cycle consists of a sequence of 6 states. Each state time lasts for two oscillator periods, thus a machine cycle takes 12 oscillator periods or 1μs if the oscillator frequency is 12MHz.

#### 12.1 Memory organization

The 83CL781/83CL782 has a 16K Program Memory (ROM) plus 256 bytes of Data Memory (RAM) on board. The device has separate address spaces for Program and Data Memory (see Figure 1). Using Ports P0 and P2, the 83CL781/83CL782 can address up to 64K bytes of external memory. The CPU generates both read and write signals (RD and WR) for external Data Memory accesses, and the read strobe (PSEN) for external Program Memory.

##### 12.1.1 Program Memory

The 83CL781/83CL782 contains 16K bytes of internal ROM. After reset the CPU begins execution at location 0000H. The lower 16K bytes of Program Memory can be implemented in either on-chip ROM or external Memory. If the EA pin is strapped to V<sub>DD</sub>, then program memory fetches from addresses 000H through 3FFFFH are directed to the internal ROM. Fetches from addresses 4000H through FFFFFH are directed to external ROM. Program counter values greater than 3FFFFH are automatically addressed to external memory regardless of the state of the EA pin.

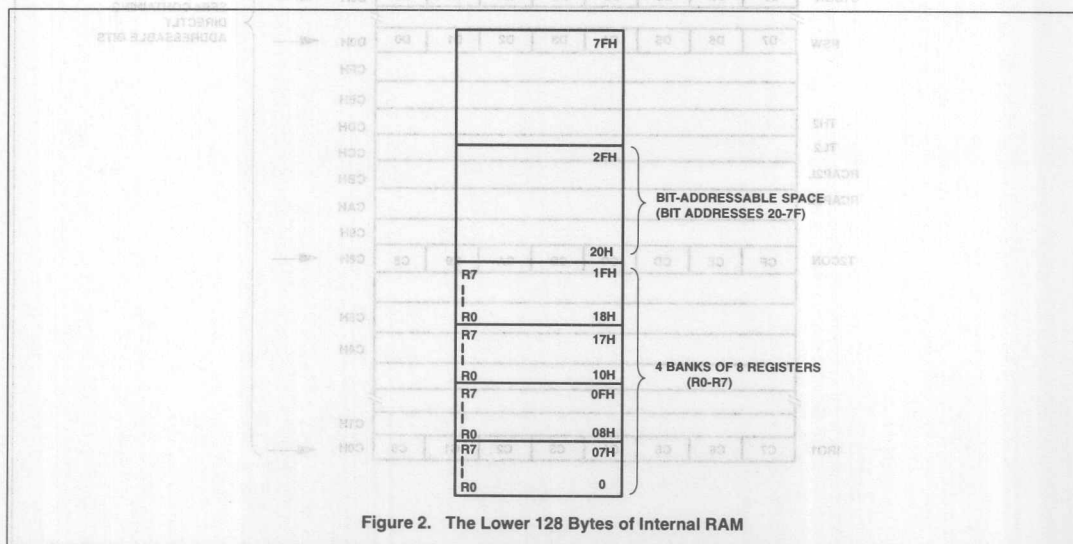
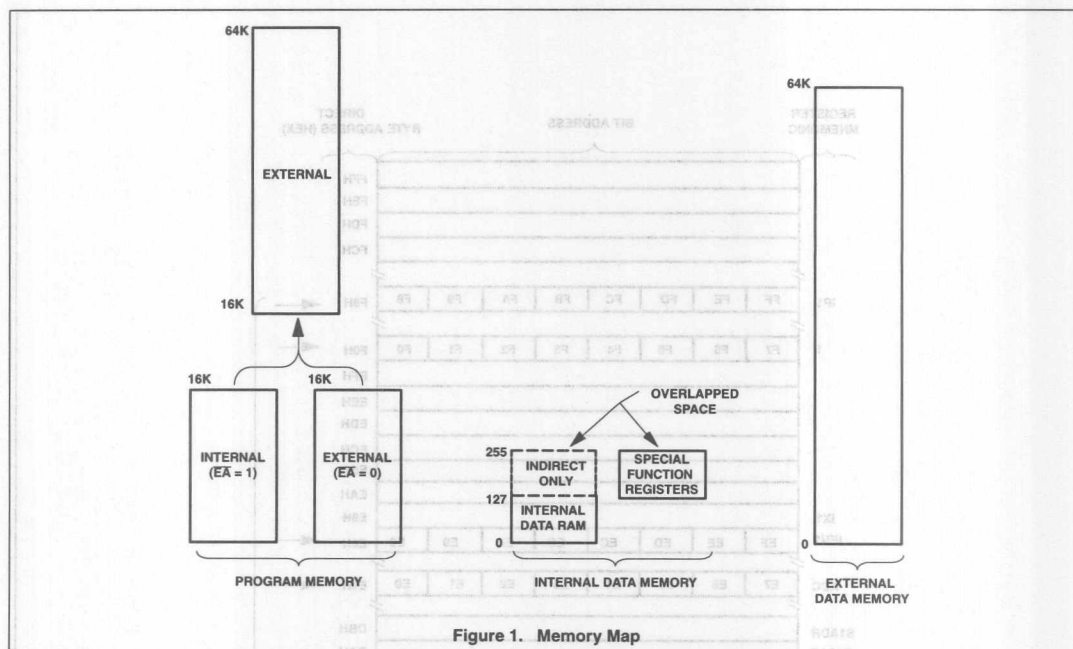
#### 12.1.2 Data Memory

The 83CL781/83CL782 contains 256 bytes of internal RAM and 34 Special Function Registers (SFR). Figure 1 shows the internal Data Memory space divided into the Lower 128, the Upper 128, and the SFR space. Internal RAM locations 0-127 are directly and indirectly addressable. Internal RAM locations 128-255 are only indirectly addressable. The special function register locations 128-255 are only directly addressable.

#### 12.1.3 Special Function Registers

The upper 128 bytes are the address locations of the SFRs. Figure 3 shows the Special Function Register (SFR) space. SFRs include the port latches, timers, peripheral control, serial I/O registers, etc. These registers can only be accessed by direct addressing. There are 128 bit addressable locations in the SFR address space (SFRs with addresses divisible by eight).

## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782



## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

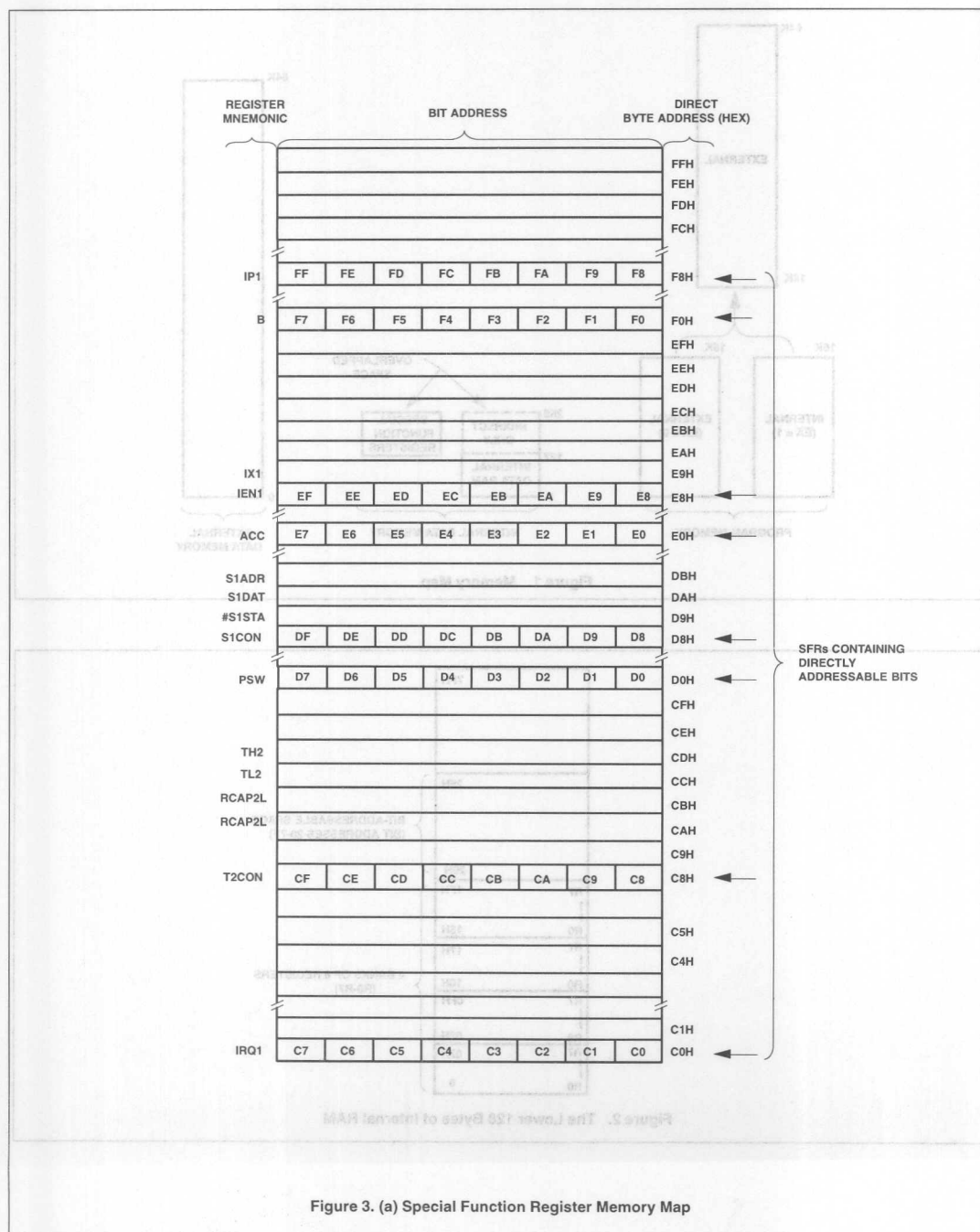
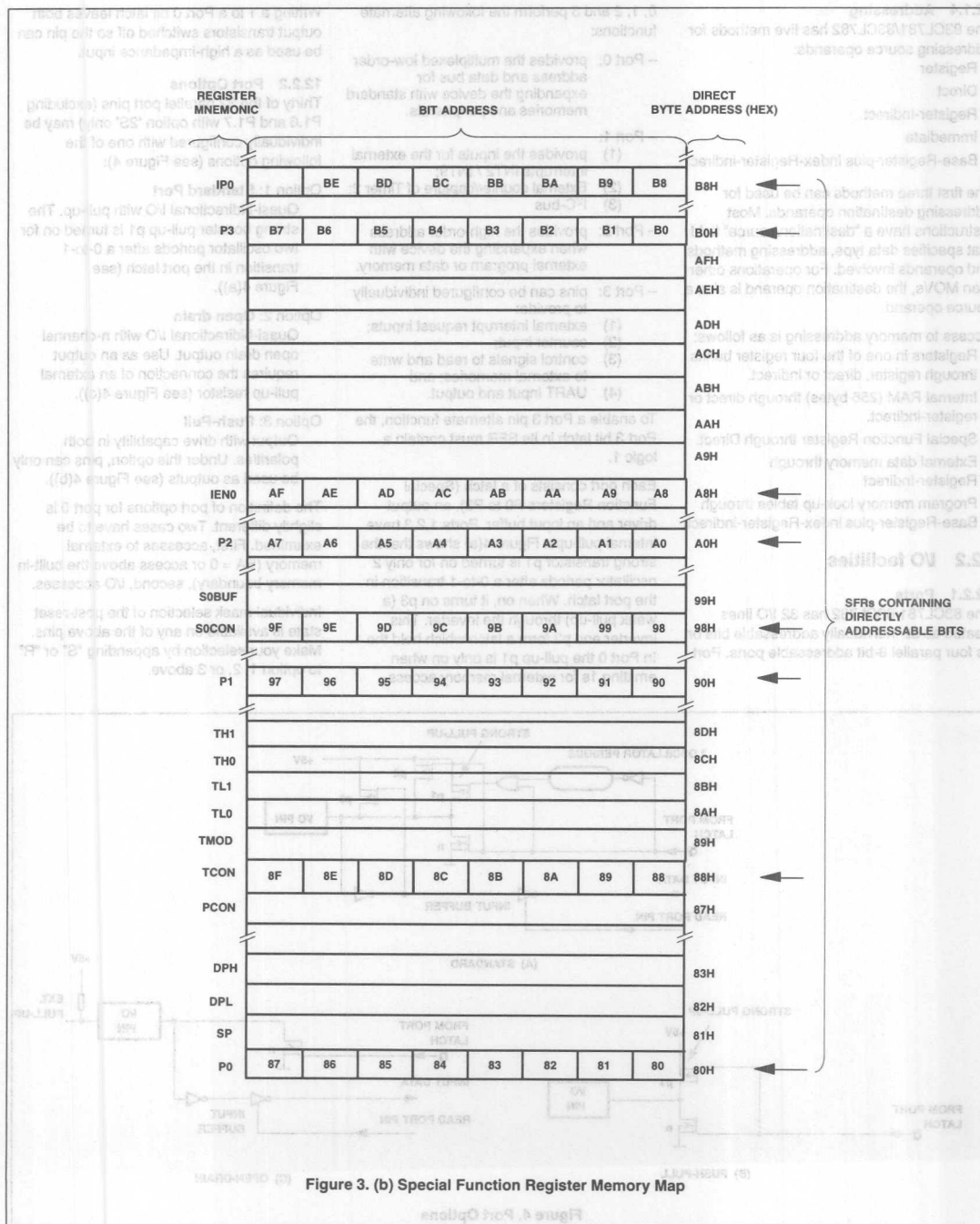


Figure 3. (a) Special Function Register Memory Map

## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782





## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

### 12.1.4 Addressing

The 83CL781/83CL782 has five methods for addressing source operands:

- Register
- Direct
- Register-Indirect
- Immediate
- Base-Register-plus Index-Register-indirect

The first three methods can be used for addressing destination operands. Most instructions have a "destination/source" field that specifies data type, addressing methods and operands involved. For operations other than MOVs, the destination operand is also a source operand.

Access to memory addressing is as follows:

- Registers in one of the four register banks through register, direct or indirect.

- Internal RAM (256 bytes) through direct or register-indirect.
- Special Function Register through Direct.
- External data memory through Register-Indirect
- Program memory look-up tables through Base-Register-plus index-Register-indirect.

## 12.2 I/O facilities

### 12.2.1 Ports

The 83CL781/83CL782 has 32 I/O lines treated as 32 individually addressable bits or as four parallel 8-bit addressable ports. Port

0, 1, 2 and 3 perform the following alternate functions:

- Port 0: provides the multiplexed low-order address and data bus for expanding the device with standard memories and peripherals.
- Port 1:
  - (1) provides the inputs for the external interrupts INT2 / INT9;
  - (2) External counter/capture of Timer 2;
  - (3) I<sup>2</sup>C-bus
- Port 2: provides the high-order address when expanding the device with external program or data memory.
- Port 3: pins can be configured individually to provide:
  - (1) external interrupt request inputs;
  - (2) counter input;
  - (3) control signals to read and write to external memories; and
  - (4) UART input and output.

To enable a Port 3 pin alternate function, the Port 3 bit latch in its SFR must contain a logic 1.

Each port consists of a latch (Special Function Registers P0 to P3), an output driver and an input buffer. Ports 1,2,3 have internal pull-ups. Figure 4(a) shows that the strong transistor p1 is turned on for only 2 oscillator periods after a 0-to-1 transition in the port latch. When on, it turns on p3 (a weak pull-up) through the inverter. This inverter and p3 form a latch which hold the 1. In Port 0 the pull-up p1 is only on when emitting 1s for external memory access.

Writing a 1 to a Port 0 bit latch leaves both output transistors switched off so the pin can be used as a high-impedance input.

### 12.2.2 Port Options

Thirty of the 32 parallel port pins (excluding P1.6 and P1.7 with option '2S' only) may be individually configured with one of the following options (see Figure 4):

#### Option 1: Standard Port

Quasi-bidirectional I/O with pull-up. The strong booster pull-up p1 is turned on for two oscillator periods after a 0-to-1 transition in the port latch (see Figure 4(a)).

#### Option 2: Open drain

Quasi-bidirectional I/O with n-channel open drain output. Use as an output requires the connection of an external pull-up resistor (see Figure 4(c)).

#### Option 3: Push-Pull

Output with drive capability in both polarities. Under this option, pins can only be used as outputs (see Figure 4(b)).

The definition of port options for port 0 is slightly different. Two cases have to be examined. First, accesses to external memory (EA = 0 or access above the built-in memory boundary), second, I/O accesses.

Individual mask selection of the post-reset state is available on any of the above pins. Make your selection by appending "S" or "R" to option 1, 2, or 3 above.

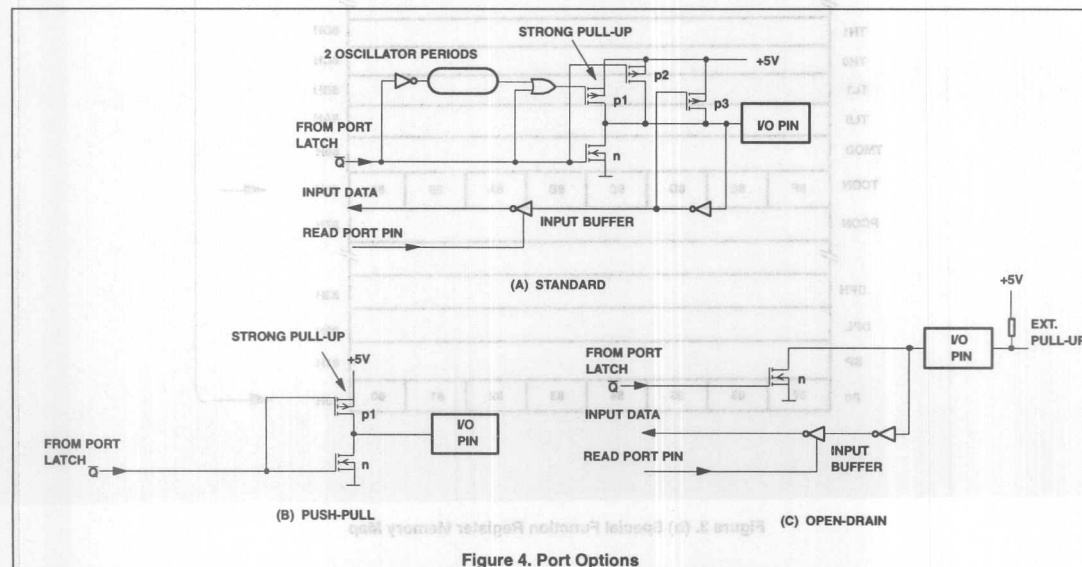


Figure 4. Port Options

# Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

## External Memory Accesses

### Option 1:

True 0 and 1 are written as address to the external memory (strong pull-up is used).

### Option 2:

An external pull-up resistor is needed for external accesses.

### Option 3:

Not allowed for external memory accesses as the port can only be used as output.

## I/O Accesses

### Option 1:

When writing a 1 to the port-latch, the strong pull-up p1 will be on for two oscillator periods. No weak pull-up exists. Without an external pull-up, this option can be used as a high-impedance input.

### Option 2:

Open drain; quasi-bidirectional I/O with n-channel open drain output. Use as an output requires the connection of an external pull-up resistor (see Figure 4(c)).

### Option 3:

Push-Pull; output with drive capability in both polarities. Under this option, pins can only be used as outputs.

Individual mask selection of the post-reset state is available on any of the above pins. Make your selection by appending "S" or "R" to option 1, 2, or 3 above.

### Option S: SET;

after reset this pin will be initialized HIGH.

### Option R: RESET;

after reset this pin will be initialized LOW.

## 12.3 Timer/event counter

The 83CL781/82CL782 contains three 16-bit Timer/Counter registers; Timer 0, Timer 1, and Timer 2 which can perform the following functions:

- Measure time intervals and pulse durations
- Count events
- Generate interrupts requests

Timer 0 and Timer 1 can be independently programmed to operate as follows:

- Mode 0: 8-bit timer or counter with divide-by-32 prescaler
- Mode 1: 16-bit time-interval or event counter
- Mode 2: 8-bit time interval or event counter with automatic reload upon overflow
- Mode 3: Timer 0 establishes TL0 and TH0 as two separate counters.

In the "Timer" function, the register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In the "Counter" function, the register is incremented in response to a 1-to-0 transition. Since it takes two machine cycles (24 oscillator periods) to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. To ensure a given level is sampled, it should be held for at least one full machine cycle.

### 12.3.1 Timer 2

Timer 2 is a 16-bit Timer/Counter. Like Timers 0 and 1, it can operate either as a timer or as an event counter. This is selected by bit C/T2 in the Special Function Register T2CON (Figure 5). It has three operating modes: "capture", "auto-load" and "baud rate generator", which are selected by bits in T2CON as shown in Table 1.

**Table 1. Timer 2 Operating Modes**

RTCLK	CP/RL2	TR2	MODE
0	0	1	16-Bit Auto-reload
0	1	1	16-Bit Capture
1	x	1	Baud Rate Generator
x	x	0	(Off)

(MSB)				(LSB)			
TF2	EXF2	GF2	RTCLK	EXEN2	TR2	C/T2	CP/RL2
<b>Symbol</b>	<b>Position</b>			<b>Name and Significance</b>			
TF2	T2CON.7			Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when RTCLK = 1.			
EXF2	T2CON.6			Timer 2 external flag set when either a capture or reload is caused by a negative transition of T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software.			
GF2	T2CON.5			General purpose flag bit.			
RTCLK	T2CON.4			Transmit clock flag. When set, causes the UART serial port to use Timer 2 overflow pulses for its receive and transmit clock in modes 1 and 3. TLCK = 0 causes Timer 1 overflows to be used for the receive and transmit clock.			
EXEN2	T2CON.3			Timer 2 external enable flag. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the UART serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.			
TR2	T2CON.2			Start/stop control for Timer 2. A logic 1 starts the timer.			
C/T2	T2CON.1			Timer or counter select. Timer 2) 0 = Internal timer (OSC/12) 1 = External event counter (falling edge triggered)			
CP/RL2	T2CON.0			Capture/Reload flag. When set, captures will occur on negative transitions at T2EX if EXEN2 = 1. When cleaned, auto-reloads will occur either with Timer 2 overflows or negative transitions at T2EX when EXEN2 = 1. When RTCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.			

**Figure 5. T2CON: Timer/Counter 2 Control Register**

# Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

In the Capture Mode there are two options which are selected by bit EXEN2 in T2CON. If EXEN2=0, then Timer 2 is a 16-bit timer or counter which upon overflowing sets bit TF2, the Timer 2 overflow bit, which can be used to generate an interrupt. If EXEN2=1, then Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input T2EX causes the current value in the Timer 2 registers, TL2 and TH2, to be captured into registers RCAP2L and RCAP2H, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set, and EXF2, like TF2, can generate an interrupt. The Capture Mode is illustrated in Figure 6.

In the auto-reload mode there are again two options, which are selected by bit EXEN2 in T2CON. If EXEN2=0, then when Timer 2 rolls

over, it not only sets TF2, but also causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2L and RCAP2H, which are preset by software. If EXEN2=1, the Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input T2EX will also trigger the 16-bit reload and set EXF2.

The auto-reload mode is illustrated in Figure 7.

The baud rate generator mode is selected by RTCLK=1. It will be described in conjunction with the UART serial port.

A conversion already in progress is aborted when the Power-down mode is entered. The result of a completed conversion (ADCI = logic 1) remains unaffected when entering the Idle mode.

## 12.4 Idle and Power-down Operation

Idle mode operation permits the interrupt, serial ports, timer blocks to continue functioning while the clock to the CPU is halted.

The following functions remain active during Idle mode. These functions may generate an interrupt or reset and thus end the Idle mode.

- Timer 0, Timer 1, Timer 2
- SIO, I<sup>2</sup>C
- External interrupt

The Power-down operation freezes the oscillator. The Power-down mode can only be activated by setting the PD bit in the PCON register.

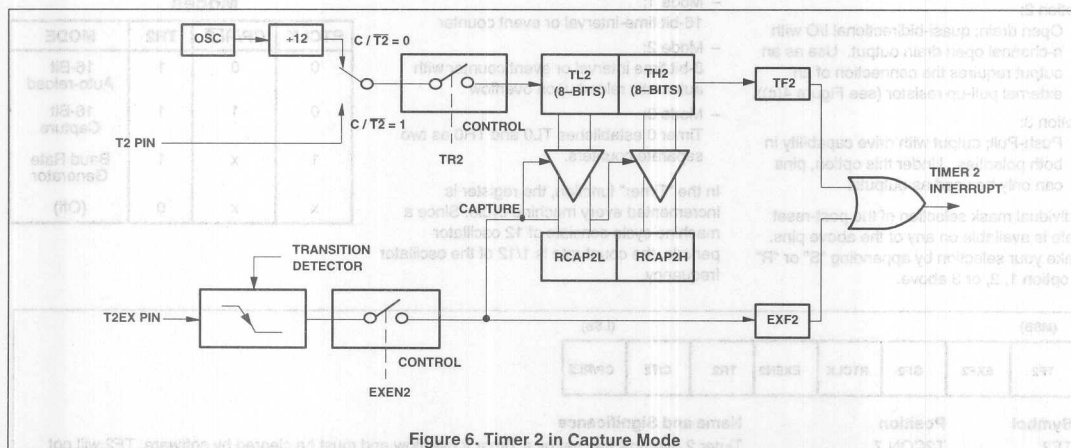


Figure 6. Timer 2 in Capture Mode

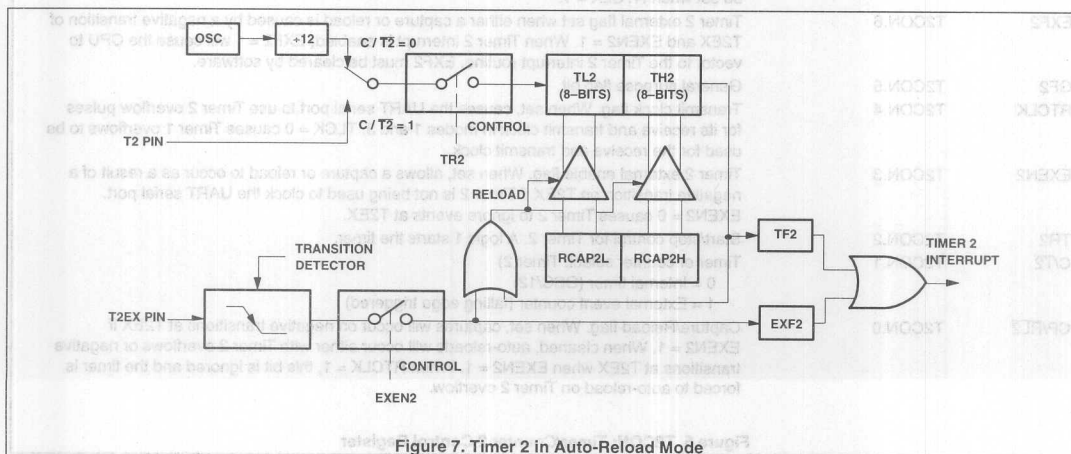


Figure 7. Timer 2 in Auto-Reload Mode

## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

### 12.4.1 Power control register (PCON)

These special modes are activated by software via the Special Function Register PCON. Its hardware address is 87H. (PCON is not bit-addressable.)

7	6	5	4	3	2	1	0
SMOD	-	-	-	GF1	GF0	PD	IDL
MSB				LSB			

BIT	SYMBOL	FUNCTION
PCON.7	SMOD	Double Baud rate bit. When set to logic 1 the baud rate is doubled when the UART serial port SIO0 is being used in 1, 2, or 3.
PCON.6	-	(reserved)
PCON.5	-	(reserved)
PCON.4	-	(reserved)
PCON.3	GF1	General-purpose flag bit
PCON.2	GF0	General-purpose flag bit
PCON.1	PD	Power-down bit. Setting this bit activates Power-down mode.
PCON.0	IDL	Idle mode bit. Setting this bit activates the Idle mode.

If logic 1s are written to PD and IDL at the same time, PD takes precedence. The reset value of PCON is (0XX00000).

### 12.4.2 Power-down mode

The instruction setting PCON.1 is the last executed prior to going into the Power-down mode. In Power-down mode the oscillator is stopped. The contents of the on-chip RAM

and SFRs are preserved. The port pins output the values held by their respective SFRs. ALE and PSEN are held LOW.

### 12.4.3 Wake-up mode

Setting the PD flag in the PCON register forces the controller into the Power-down mode. Setting this flag enables the controller to be awakened from the Power-down mode with either the external interrupts INT2/INT9, or a reset operation.

The wake-up operation after power-down in this controller has two basic approaches:

#### 12.4.3.1 Wake-up using INT2 to INT9

If INT2 to INT9 are enabled, the 83CL781/83CL782 can be awakened from power-down mode with the external interrupts. To ensure that the oscillator is stable before the controller restarts, the internal clock will remain inactive for 1536 oscillator periods. This is controlled by an on-chip delay counter.

#### 12.4.3.2 Wake-up using RESET

To wake-up the 83CL781/83CL782 the RESET pin has to be kept HIGH for a minimum of 24 periods. The on-chip delay counter is inactive. The user has to ensure that the oscillator is stable before any operation is attempted. Figure 8 illustrates the two possibilities for wake-up.

### 12.4.4 Idle mode

The instruction that sets PCON.0 is the last instruction executed in the normal operating mode before Idle mode is activated. Once in the Idle mode, the CPU status is preserved in its entirety: the Stack Pointer, Program Counter, Program Status Word, Accumulator, RAM and all other registers maintain their data during Idle mode. The status of the external pins during Idle mode is shown in Table 2.

There are two methods used to terminate the Idle mode. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, terminating Idle mode. The interrupt is serviced, and following the return-from-interrupt instruction RETI, the next instruction to be executed will be the one which follows the instruction that wrote a logic 1 to PCON.0.

Flag bits GF0 and GF1 may be used to determine whether the interrupt was received during normal execution or during Idle mode. For example, the instruction that writes to PCON.0 can also set or clear one or both flag bits. When Idle mode is terminated by an interrupt, the service routine can examine the status of the flag bits.

The second method of terminating the Idle mode is with an external hardware reset, or an internal reset caused by an overflow of Timer T3. Since the oscillator is still running, the hardware reset is required to be active for two machine cycles (24 oscillator periods) to complete the reset operation.

Reset redefines all SFRs, but does not affect the on-chip RAM.

In the Power-down mode,  $V_{DD}$  may be reduced to minimize power consumption. However, the supply voltage must not be reduced until Power-down mode is active, and must be held active until the oscillator has restarted and stabilized.

The status of the external pins during Idle and Power-down mode is shown in Table 2. If the Power-down mode is activated whilst accessing external memory, port data held in the Special Function Register P2 is restored to Port 2. If the data is a logic 1, the port pin is held HIGH during the Power-down mode by the strong pull-up transistor p1 (see Figure 4(a)).

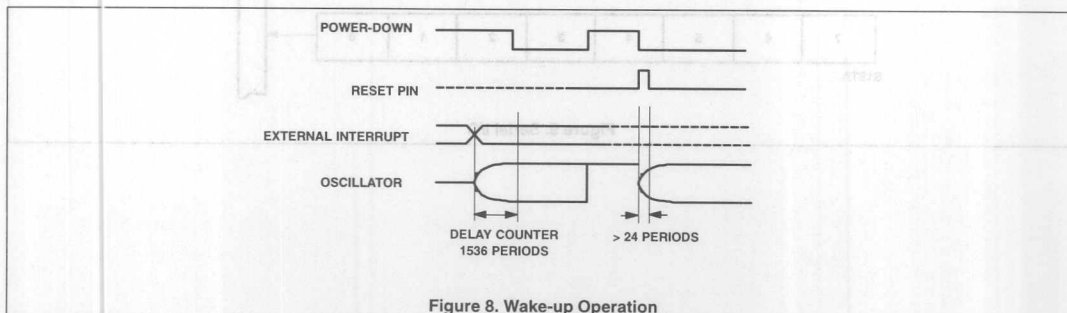


Figure 8. Wake-up Operation

Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

**Table 2. Status of the External Pins during Idle and Power-down Modes**

MODE	MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle (1)	Internal	1	1	Port data	Port data	Port data	Port data
Idle (1)	External	1	1	Floating	Port data	Address	Port data
Power-down	Internal	0	0	Port data	Port data	Port data	Port data
Power-down	External	0	0	Floating	Port data	Port data	Port data

## 12.5 I<sup>2</sup>C Bus Serial I/O

The serial port supports the twin line I<sup>2</sup>C-bus. The I<sup>2</sup>C bus consists of a data line (SDA) and a clock line (SCL). These lines also function as I/O port lines P1.7 and P1.6 respectively. The system is unique because data transport, clock generation, address recognition and bus control arbitration are all

controlled by hardware. The I<sup>2</sup>C bus serial I/O has complete autonomy in byte handling and operates in four modes:

- Master transmitter
- Master receiver
- Slave transmitter
- Slave receiver

These functions are controlled by the S1CON register. S1STA is the status register whose contents may also be used as a vector to various service routines. S1DAT is the data shift register and S1ADR the slave address register. Slave address recognition is performed by hardware.

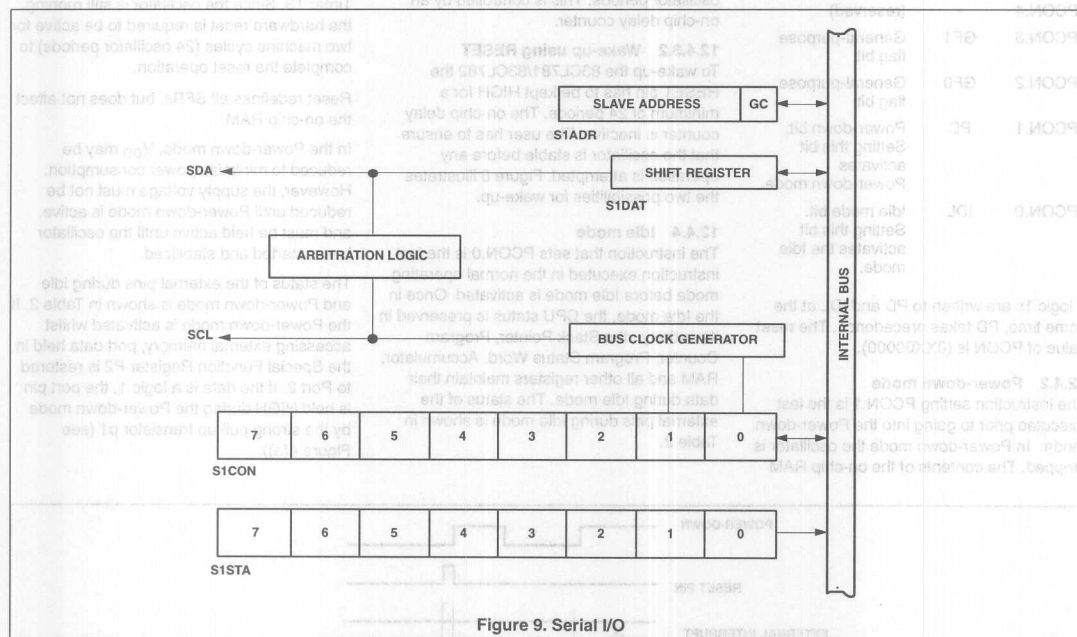


Figure 9. Serial I/O



## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

## Serial Control Register S1CON (D8H)

CR2	ENS1	STA	STO	SI	AA	CR1	CR0
-----	------	-----	-----	----	----	-----	-----

**CR0, CR1, CR2** These three bits determine the serial clock frequency when SIO is in a master mode. See Table 3.

**AA** Assert Acknowledge bit. When the AA flag is set, an acknowledge (low level SDA) will be returned during the acknowledge clock pulse on the SCL line when:

- own slave address is received
- general call address is received (S1ADR.0=1)
- data byte received while device is programmed as master
- data byte received while device is a selected slave

With AA=0, no acknowledge will be returned. Consequently, no interrupt is requested when the "own slave address" or general call address is received.

**SI** SIO interrupt flag. When the SI flag is set, an acknowledge is returned after any one of the following conditions:

- a start condition is generated in master mode
- own slave address received during AA=1
- general call address received while S1ADR.0 and AA=1
- data byte received or transmitted in master mode (even if arbitration is lost)
- data byte received or transmitted as selected slave
- stop or start condition received as selected slave receiver or transmitter

**STO** STOP flag. With this bit set while in master mode a STOP condition is generated. When a STOP condition is detected on the bus, the SIO hardware clears the STO flag. In the slave mode, the STO flag may also be set to recover from an error condition. In this case, no STOP condition is transmitted to the I<sup>2</sup>C bus. However, the SIO hardware behaves as if a STOP condition has been received and releases SDA and SCL. The SIO then switches to the "not addressed" receiver mode. The STO flag is automatically cleared by hardware.

**STA** START flag. When the STA bit is set in slave mode, the SIO hardware checks the status of the I<sup>2</sup>C bus and generates a START condition if the bus is free. If STA is set while the SIO is in master mode, SIO transmits a repeated START condition.

**ENS1** When ENS1=0, the SIO is disabled. The SDA and SCL outputs are in a high-impedance state; P1.6 and P1.7 function as open drain ports. When ENS=1, the SIO is enabled. The P1.6 and P1.7 port latches must be set to logic 1.

Table 3. SCL Frequency

CR2	CR1	CR0	f <sub>osc</sub> Divided By	Bit Rate (kHz) at f <sub>osc</sub>		
				3.58MHz	6MHz	12MHz
0	0	0	256	14.0	23.4	46.9
0	0	1	224	16.0	26.8	53.6
0	1	0	192	18.6	31.3	62.5
0	1	1	160	22.4	37.5	75.0
1	0	0	960	3.73	6.25	12.5
1	0	1	120	29.8	50	100
1	1	0	60	59.7	100	—
1	1	1	not allowed	—	—	—

## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

## Status Register S1STA (D9H)

SC4	SC3	SC2	SC1	SC0	0	0	0
-----	-----	-----	-----	-----	---	---	---

S1STA is an 8-bit read-only special function register. S1STA.3-S1STA.7 hold a status code. S1STA.0-S1STA.2 are held LOW. The contents of S1STA may be used as a vector to a service routine. This optimizes response time of the software and consequently that of the I<sup>2</sup>C bus.

The following is a list of the status code:

## Abbreviations used:

SLA:	7-bit slave address
R:	Read bit
W:	Write bit
ACK:	Acknowledgement (acknowledge bit = 0)
ACKNOT:	Not Acknowledge (acknowledge bit = 1)
DATA:	8-bit byte to or from the I <sup>2</sup> C bus
MST:	Master
SLV:	Slave
TRX:	Transmitter
REC:	Receiver

## MST/TRX Mode

## S1STA Value

08H	A START condition has been transmitted
10H	A repeated START condition has been transmitted
18H	SLA and W have been transmitted, ACKNOT received
28H	DATA of S1DAT has been transmitted, ACK received
30H	DATA of S1DAT has been transmitted, ACKNOT received
38H	Arbitration lost in SLA, R/W or DATA

## MST/REC Mode

## S1STA Value

38H	Arbitration lost while returning ACKNOT
40H	SLA and R have been transmitted, ACK received
48H	SLA and R have been transmitted, ACKNOT received
50H	DATA has been received, ACK returned
58H	DATA has been received, ACKNOT returned

## SLV/REC Mode

## S1STA Value

60H	Own SLA and W have been received, ACK returned
68H	Arbitration lost in SLA, RW as MST. Own SLA and W have been received, ACK returned.
70H	General Call has been received, ACK returned
78H	Arbitration lost in SLA, RW as MST. General Call has been received.
80H	Previously addressed with own SLA. DATA byte received, ACK returned.
88H	Previously addressed with General Call. DATA byte has been received, ACKNOT has been returned.
90H	Previously addressed with General Call. DATA byte has been received, ACK has been returned.
98H	Previously addressed with General Call. DATA byte has been received, ACKNOT has been returned.
A0H	A STOP condition or repeated START condition has been received while still addressed as SLV/REC or SLV/TRX.

## SLV/TRX Mode

## S1STA Value

A8H	Own SLA and R have been received, ACK returned
B0H	Arbitration lost in SLA, R/W as MST. Own SLA and R have been received, ACK returned.
B8H	DATA byte has been transmitted, ACK received.
C0H	DATA byte has been transmitted, ACKNOT received.

## Miscellaneous

## S1STA Value

00H	Bus error during MST mode or selected SLV mode, due to an erroneous START or STOP condition.
-----	--

## Data Shift Register S1DAT (DAH)

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

This register contains the serial data to be transmitted or data that has just been received. Bit 7 is transmitted or received first; i.e., data is shifted from left to right.

## Own Address Register S1ADR (DBH)

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

This 8-bit register may be loaded with the 7-bit address to which the controller will respond when programmed as a slave receiver/transmitter. The LSB bit (GC) is used to determine whether the general CALL address is recognized.

S1ADR.0, GC: 0 = general CALL address is not recognized.  
1 = general CALL is recognized.

S1ADR.7 - 1: own slave address

## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

### 12.6 Standard Serial Interface SIQ0: UART

This serial port is full duplex, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the register. However, if the first byte still hasn't been read by the time reception of the second byte is complete, one of the bytes will be lost). The serial port receive and transmit registers are both accessed at Special Function Register S0BUF. Writing to S0BUF loads the transmit register, and reading S0BUF accesses a physically separate receive register.

The serial port can operate in 4 modes:

**Mode 0:** Serial data enters and exits through RxD. TxD outputs the shift clock. 8 bits are transmitted/received (LSB first). The baud is fixed at 1/12 the oscillator frequency.

**Mode 1:** 10 bits are transmitted (through TxD) or received (through RxD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in Special Function Register SCON. The baud rate is variable.

**Mode 2:** 11 bits are transmitted (through TxD) or received (through RxD): start bit (0), 8 data bits (LSB first), a programmable 9th

data bit, and a stop bit (1). On Transmit, the 9th data bit (TB8 in SCON) can be assigned the value of 0 or 1. Or, for example, the parity bit (P, in the PSW) could be moved into TB8. On receive, the 9th data bit goes into RB8 in Special Function Register SCON, while the stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 the oscillator frequency.

**Mode 3:** 11 bits are transmitted (through TxD) or received (through RxD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit and a stop bit (1). In fact, Mode 3 is the same as Mode 2 in all respects except baud rate. The baud rate in Mode 3 is variable.

In all four modes, transmission is initiated by any instruction that uses S0BUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit if REN = 1.

#### 12.6.1 Multiprocessor Communications

Modes 2 and 3 have a special provision for multiprocessor communications. In these modes, 9 data bits are received. The 9th one goes into RB8. Then comes a stop bit. The port can be programmed such that when the stop bit is received, the serial port interrupt will be activated only if RB8 = 1. This feature is enabled by setting bit SM2 in SCON. A

way to use this feature in multiprocessor systems is as follows:

When the master processor wants to transmit a block of data to one of several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte. With SM2 = 1, no slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave will clear its SM2 bit and prepare to receive the data bytes that will be coming. The slaves that weren't being addressed leave their SM2s set and go on about their business, ignoring the coming data bytes.

SM2 has no effect in Mode 0, and in Mode 1 can be used to check the validity of the stop bit. In a Mode 1 reception, if SM2 = 1, the receive interrupt will not be activated unless a valid stop bit is received.

#### 12.6.2 Serial port control register

The serial port control and status register is the Special Function Register S0CON, shown in Figure 10. The register contains not only the mode selection bits, but also the 9th data bit for transmit and receive (TB8 and RB8), and the serial port interrupt bits (TI and RI).

MSB				LSB			
SM0	SM1	SM2	REN	TB8	RB8	TI	RI
Where SM0, SM1 specify the serial port mode, as follows:							
SM0	SM1	MODE	DESCRIPTION	BAUD RATE			
0	0	0	Shift register	$f_{osc} / 12$			
0	1	1	8-bit UART	variable			
1	0	2	9-bit UART	$f_{osc} / 64$ or $f_{osc} / 32$			
1	1	3	9-bit variable UART				
<b>SM2</b>	enables the multiprocessor communication feature in Modes 2 and 3. In Mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received ninth data bit (RB8) is 0. In Mode 1, if SM2=1 then RI will not be activated if a valid stopbit was not received. In Mode 0, SM2 should be 0.						
<b>REN</b>	enables serial reception. Set by software to enable reception. Clear by software to disable reception.						
<b>TB8</b>	is the ninth data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired.						
<b>RB8</b>	in Modes 2 and 3, is the ninth data bit that was received. In Mode 1, if SM2=0, RB8 is the stop bit that was received. In Mode 0, RB8 is not used.						
<b>TI</b>	is transmit interrupt flag. Set by hardware at the end of the eighth bit time in Mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.						
<b>RI</b>	is receive interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM2). Must be cleared by software.						

**Figure 10. Serial Port Control (SCON) Register**

Figure 10. Serial Port Control (SCON) Register

## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

### Baud Rates

The baud rate in Mode 0 is fixed: Mode 0 Baud Rate = Oscillator Frequency / 12. The baud rate in Mode 2 depends on the value of bit SMOD in Special Function Register PCON. If SMOD = 0 (which is the value on reset), the baud rate is 1/64 the oscillator frequency. If SMOD = 1, the baud rate is 1/32 the oscillator frequency.

Mode 2 Baud Rate =  $(2^{\text{SMOD}} / 64) \times (\text{Oscillator Frequency})$

The baud rates in Modes 1 and 3 are determined by the Timer 1 or Timer 2 overflow rate.

### Using Timer 1 to Generate Baud Rates

When Timer 1 is used as the baud rate generator, the baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate and the value of SMOD as follows:

Modes 1,3 Baud Rate =  $(2^{\text{SMOD}} / 32) \times (\text{Timer 1 Overflow Rate})$

The Timer 1 interrupt should be disabled in this application. The Timer itself can be configured for either "timer" or "counter" operation, and in any of its 3 running modes. In most typical applications, it is configured for "timer" operation, in the auto-reload mode (high nibble of TMOD = 0010B). In that case the baud rate is given by the formula:

Mode 1, 3 Baud Rate =  $\{(2^{\text{SMOD}} / 32) \times (\text{Oscillator Frequency})\} / (12 \times (256 - (\text{TH1})))$

One can achieve very low baud rates with Timer 1 by leaving the Timer 1 interrupt

enabled, and configuring the Timer to run as a 16-bit timer (high nibble of TMOD = 0001B), and using the Timer 1 interrupt to do a 16-bit software reload. Table 4 lists various commonly used baud rates and how they can be obtained from Timer 1.

### Using Timer 2 to Generate Baud Rates

Timer 2 is selected as the baud rate generator by setting RTCLK in T2CON (Figure 11). Setting RTCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 11.

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by software.

Now, the baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate as follows:

Modes 1,3 Baud Rate =  $(\text{Timer 2 Overflow Rate}) / 16$

The Timer can be configured for either "timer" or "counter" operation. In the most typical applications, it is configured for "timer" operation (C/T2 = 0). "Timer" operation is a little different for Timer 2 when it's being used as a baud rate generator. Normally, as a timer it would increment every machine cycle (thus at 1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time (thus at 1/2 the oscillator

frequency). In that case the baud rate is given by the formula:

Modes 1,3 Baud Rate =  $(\text{Oscillator Frequency}) / \{32 \times (65536 - (\text{RCAP2H}, \text{RCAP2L}))\}$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

Timer 2 as a baud rate generator is shown in Figure 11. This Figure is valid only if RTCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2, and will not generate an interrupt. Therefore, the Timer 2 interrupt does not have to be disabled when Timer 2 is in the baud rate generator mode. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt, if desired.

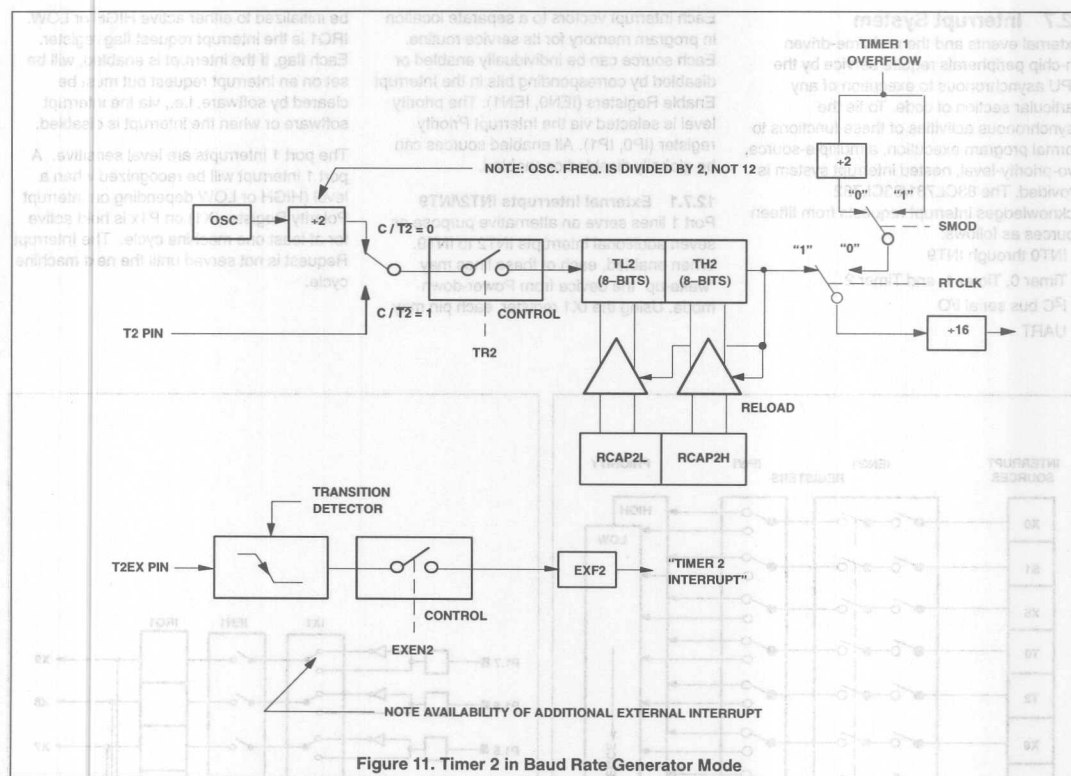
It should be noted that when Timer 2 is running (TR2 = 1) in "timer" function in the baud rate generator mode, one should not try to read or write TH2 or TL2. Under these conditions the Timer is being incremented every state time, and the results of a read or write may not be accurate. The RCAP registers may be read, but shouldn't be written to, because a write might overlap a reload and cause write and/or reload errors. Turn the Timer off (clear TR2) before accessing the Timer 2 or RCAP register, in this case.

IR	IT	URR	TA	MR	NR	SM2	SM1	SM0
----	----	-----	----	----	----	-----	-----	-----

Table 4. Timer 1 Generated Commonly Used Baud Rates

BAUD RATE	f <sub>osc</sub> (MHz)	SMOD	C/T	TIMER 1 MODE	RELOAD VALUE
Mode 0 Max: 1MHz	12	X	X	X	X
Mode 2 Max: 375K	12	1	X	X	X
Mode 1, 3: 62.5K	12	1	0	2	FFH
19.2K	11.059	1	0	2	FDH
9.6K	11.059	0	0	2	FDH
4.8K	11.059	0	0	2	FAH
2.4K	11.059	0	0	2	F4H
1.2K	11.059	0	0	2	E8H
137.5K	11.986	0	0	2	1DH
110	6	0	0	2	72H
110	12	0	0	1	FEEDH

## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782





## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

### 12.7 Interrupt System

External events and the real-time-driven on-chip peripherals require service by the CPU asynchronous to execution of any particular section of code. To tie the asynchronous activities of these functions to normal program execution, a multiple-source, two-priority-level, nested interrupt system is provided. The 83CL781/83CL782

acknowledges interrupt requests from fifteen sources as follows:

- INT0 through INT9
- Timer 0, Timer 1, and Timer 2
- I<sup>2</sup>C bus serial I/O
- UART

Each interrupt vectors to a separate location in program memory for its service routine. Each source can be individually enabled or disabled by corresponding bits in the Interrupt Enable Registers (IEN0, IEN1). The priority level is selected via the Interrupt Priority register (IP0, IP1). All enabled sources can be globally disabled or enabled.

#### 12.7.1 External Interrupts INT2/INT9

Port 1 lines serve an alternative purpose as seven additional interrupts INT2 to INT9. When enabled, each of these lines may "wake-up" the device from Power-down mode. Using the IX1 register, each pin may

be initialized to either active HIGH or LOW. IRQ1 is the interrupt request flag register. Each flag, if the interrupt is enabled, will be set on an interrupt request but must be cleared by software, i.e., via the interrupt software or when the interrupt is disabled.

The port 1 interrupts are level sensitive. A port 1 interrupt will be recognized when a level (HIGH or LOW depending on Interrupt Polarity Register IX1) on P1x is held active for at least one machine cycle. The Interrupt Request is not served until the next machine cycle.

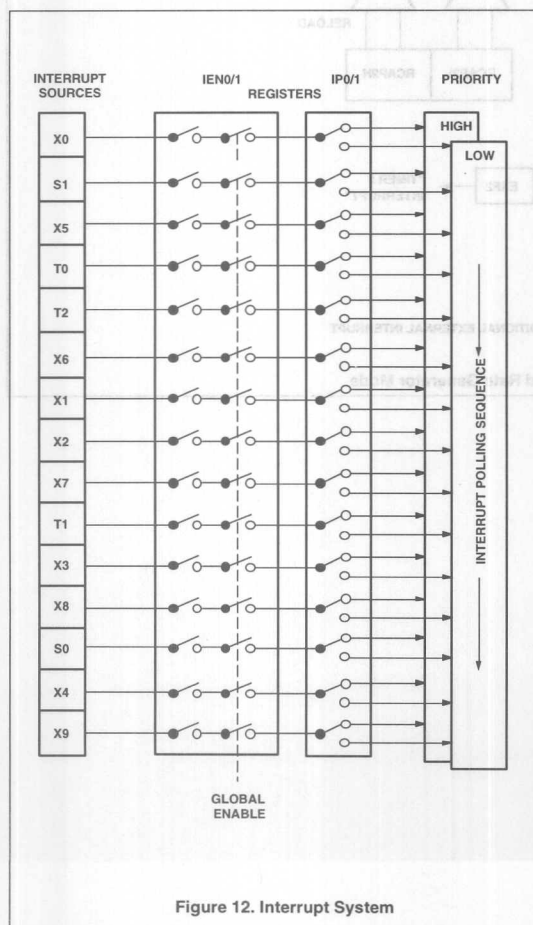


Figure 12. Interrupt System

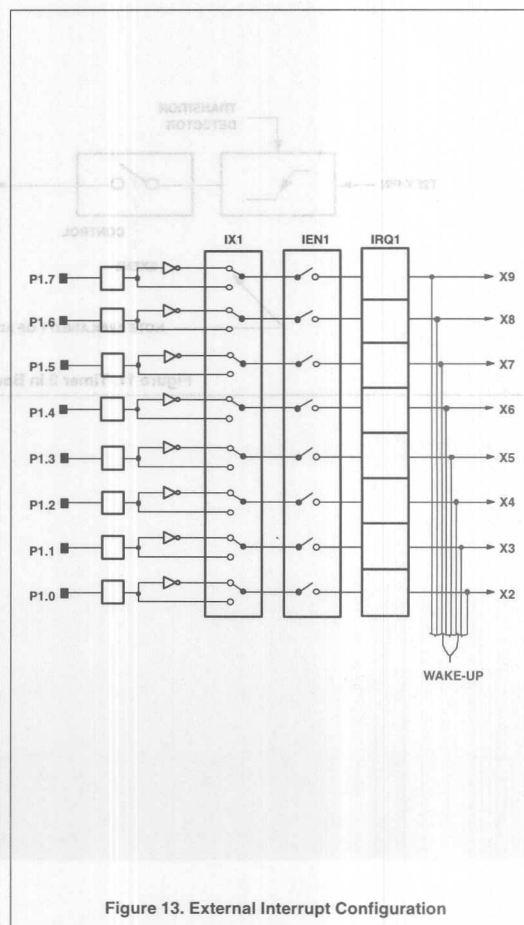


Figure 13. External Interrupt Configuration

## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

## Interrupt Enable Register

## IEN0, IEN1

## IEN0 (A8H)

EA	ET2	ES1	ES0	ET1	EX1	ET0	EX0
----	-----	-----	-----	-----	-----	-----	-----

BIT	SYMBOL	FUNCTION
IEN0.7	EA	General enable/disable control 0 = no interrupt is enabled; 1 = any individually enabled interrupt will be accepted
IEN0.6	ET2	Enable T2 interrupt
IEN0.5	ES1	Enable I <sup>2</sup> C interrupt
IEN0.4	ES0	Enable UART SIO interrupt
IEN0.3	ET1	Enable Timer T1 interrupt
IEN0.2	EX1	Enable external interrupt 1
IEN0.1	ET0	Enable Timer T0 interrupt
IEN0.0	EX0	Enable external interrupt 0

## IEN1 (E8H)

EX9	EX8	EX7	EX6	EX5	EX4	EX3	EX2
-----	-----	-----	-----	-----	-----	-----	-----

BIT	SYMBOL	FUNCTION
IEN1.7	EX9	Enable external interrupt 9
IEN1.6	EX8	Enable external interrupt 8
IEN1.5	EX7	Enable external interrupt 7
IEN1.4	EX6	Enable external interrupt 6
IEN1.3	EX5	Enable external interrupt 5
IEN1.2	EX4	Enable external interrupt 4
IEN1.1	EX3	Enable external interrupt 3
IEN1.0	EX2	Enable external interrupt 2

where 0 = interrupt disabled  
1 = interrupt enabled

## Interrupt Priority Register

## IP0, IP1

## IP0 (B8H)

-	PT2	PS1	PS0	PT1	PX1	PT0	PX0
---	-----	-----	-----	-----	-----	-----	-----

BIT	SYMBOL	FUNCTION
IP0.7	-	Unused
IP0.6	PT2	Timer 2 interrupt priority level
IP0.5	PS1	I <sup>2</sup> C interrupt priority level
IP0.4	PS0	UART SIO interrupt priority level
IP0.3	PT1	Timer 1 interrupt priority level
IP0.2	PX1	External interrupt 1 priority level
IP0.1	PT0	Timer 0 interrupt priority level
IP0.0	PX0	External interrupt 0 priority level

## IP1 (F8H)

PX9	PX8	PX7	PX6	PX5	PX4	PX3	PX2
-----	-----	-----	-----	-----	-----	-----	-----

BIT	SYMBOL	FUNCTION
IP1.7	PX9	External interrupt 9 priority level
IP1.6	PX8	External interrupt 8 priority level
IP1.5	PX7	External interrupt 7 priority level
IP1.4	PX6	External interrupt 6 priority level
IP1.3	PX5	External interrupt 5 priority level
IP1.2	PX4	External interrupt 4 priority level
IP1.1	PX3	External interrupt 3 priority level
IP1.0	PX2	External interrupt 2 priority level

Interrupt priority is as follows:  
0 = low priority  
1 = high priority

## Interrupt Polarity Register

## IX1

## IX1 (E9H)

IL9	IL8	IL7	IL6	IL5	IL4	IL3	IL2
-----	-----	-----	-----	-----	-----	-----	-----

Writing either a "1" or "0" to an IX1 register bit sets the polarity level of the corresponding external interrupt to active HIGH or LOW respectively.

BIT	SYMBOL	FUNCTION
IX1.7	IL9	External interrupt 9 polarity level
IX1.6	IL8	External interrupt 8 polarity level
IX1.5	IL7	External interrupt 7 polarity level
IX1.4	IL6	External interrupt 6 polarity level
IX1.3	IL5	External interrupt 5 polarity level
IX1.2	IL4	External interrupt 4 polarity level
IX1.1	IL3	External interrupt 3 polarity level
IX1.0	IL2	External interrupt 2 polarity level

## Interrupt Request Flag Register IRQ1

## IRQ1 (C0H)

IQ9	IQ8	IQ7	IQ6	IQ5	IQ4	IQ3	IQ2
-----	-----	-----	-----	-----	-----	-----	-----

BIT	SYMBOL	FUNCTION
IRQ1.7	IQ9	External interrupt 9 request flag
IRQ1.6	IQ8	External interrupt 8 request flag
IRQ1.5	IQ7	External interrupt 7 request flag
IRQ1.4	IQ6	External interrupt 6 request flag
IRQ1.3	IQ5	External interrupt 5 request flag
IRQ1.2	IQ4	External interrupt 4 request flag
IRQ1.1	IQ3	External interrupt 3 request flag
IRQ1.0	IQ2	External interrupt 2 request flag

## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

### 12.7.2 Interrupt vectors

(highest)	Vector	Source
X0	0003H	external/0
S1	002BH	I <sup>2</sup> C port
X5	0053H	external 5
T0	000BH	timer 0
T2	0033H	timer 2
X6	005BH	external 6
X1	0013H	external 1
X2	003BH	external 2
X7	0063H	external 7
T1	001BH	timer 1
X3	0043H	external 3
X8	006BH	external 8
S0	0023H	UART
X4	004BH	external 4
X9	0073H	external 9
(lowest)		

#### 12.7.2.1 Interrupt Priority

Each interrupt priority source can be set to either high or low priority. If both priorities are requested simultaneously, the controller will branch to the high priority vector.

A low priority interrupt can only be interrupted by a high priority interrupt. A high priority interrupt routine cannot be interrupted.

### 12.7.3 Related registers

The following registers are used in conjunction with the interrupt system:

REGISTER	FUNCTION	SFR ADDRESS
IX1	Interrupt polarity register	E9H
IRQ1	Interrupt request flag register	C0H
IEN0	Interrupt enable register	A8H
IEN1	Interrupt enable register (INT2-INT9)	E8H
IP0	Interrupt priority register	B8H
IP1	Interrupt priority register (INT2-INT9)	F8H

### 12.8 Oscillator Circuitry

The on-chip oscillator circuitry of the 83CL781/83CL782 is a single-stage inverting amplifier biased by an internal feedback resistor (Figure 14). For operation as a standard quartz oscillator, no external components are needed (except at 32kHz). When using external capacitors, ceramic resonators, coils and RC networks to drive the oscillator, five different configurations are supported (see Figure 15 and oscillator options).

In the Power-down mode the oscillator is stopped and XTAL1 is pulled HIGH. The oscillator inverter is switched off to ensure no current will flow regardless of the voltage at XTAL1. To drive the device with an external clock source, apply the external clock signal to XTAL1, and leave XTAL2 to float, as shown in Figure 15(f). There are no requirements on the duty cycle of the external clock, since the input to the internal clocking circuitry is buffered by a flip-flop.

The following options are provided for optimum on-chip oscillator performance. Please state option when ordering.

#### 12.8.1 Oscillator options (see Figure 15)

Osc.1: Figure 15(c). An option for 32kHz clock applications with external trimmer for frequency adjustment. A 4.7M $\Omega$  bias resistor is needed for use in parallel with the crystal.

Osc.2: Figure 15(e). An option for low-power, low-frequency operations using LC components.

Osc.3: An option for medium frequency range applications.

Osc.4: An option for high frequency range applications.

RC: Figure 15(g). An option for an RC oscillator.

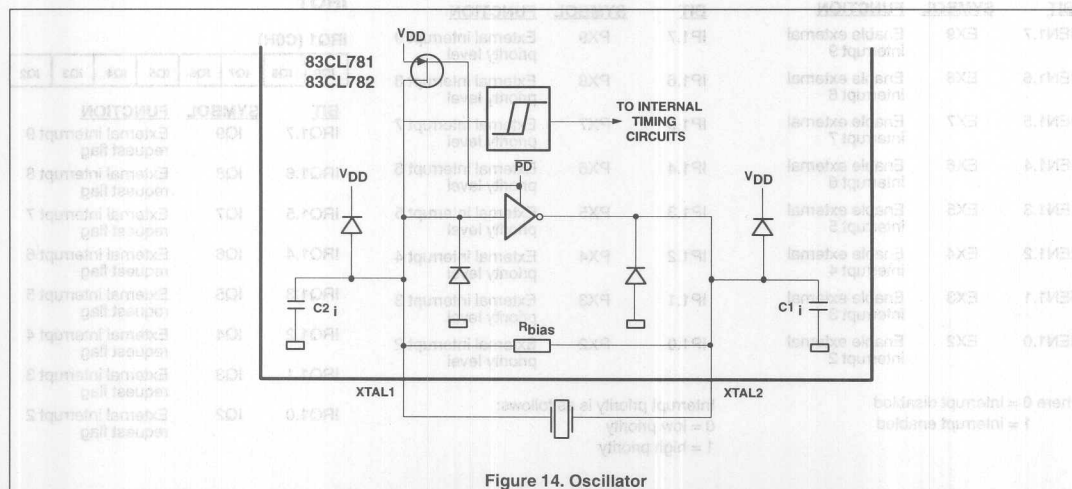


Figure 14. Oscillator

## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

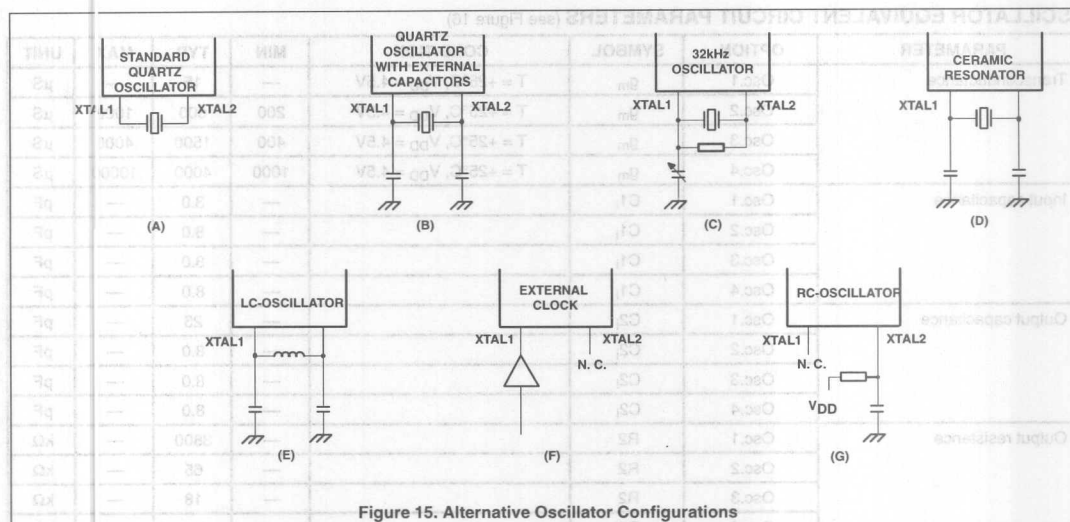


Figure 15. Alternative Oscillator Configurations

## 12.8.2 Oscillator Type Selection Guide

## OSCILLATOR TYPE SELECTION GUIDE

RESONATOR	f (MHz)	OPTION	C1 EXT. (pF)		C2 EXT. (pF)		MAX. RESONATOR SERIES RESISTANCE 15kΩ <sup>1</sup>
			MIN	MAX	MIN	MAX	
Quartz	0.032	Osc.1	0	0	5	15	600Ω
Quartz	1.0	Osc.2	0	30	0	30	100Ω
Quartz	3.58	Osc.2	0	15	0	15	75Ω
Quartz	4.0	Osc.2	0	20	0	20	60Ω
Quartz	6.0	Osc.3	0	10	0	10	60Ω
Quartz	10.0	Osc.4	0	15	0	15	40Ω
Quartz	12.0	Osc.4	0	10	0	10	20Ω
PXE	0.455	Osc.2	40	50	40	50	100Ω
PXE	1.0	Osc.2	15	50	15	50	10Ω
PXE	3.58	Osc.2	0	40	0	40	10Ω
PXE	4.0	Osc.2	0	40	0	40	5Ω
PXE	6.0	Osc.2	0	20	0	20	6Ω
PXE	10.0	Osc.3	0	15	0	15	6Ω
PXE	12.0	Osc.4	10	40	10	40	10μH = 1Ω
LC		Osc.2	20	90	20	90	100μH = 5Ω 1mH = 75Ω

## NOTE:

1. 32kHz quartz crystals with a series resistance higher than 15kΩ will reduce the guaranteed supply voltage range to 2.5 - 3.5V.

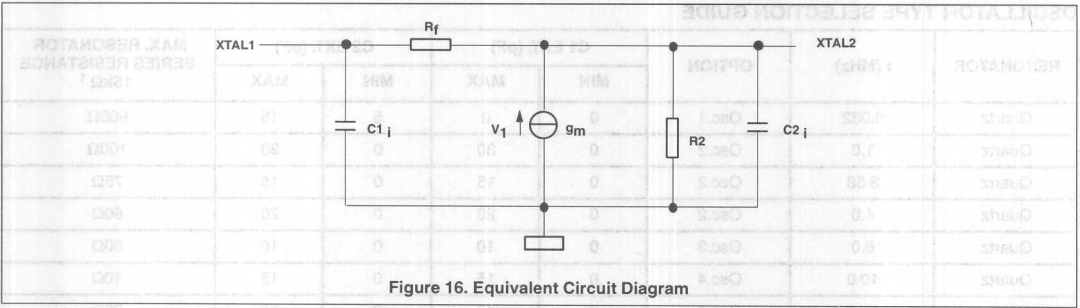
Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

OSCILLATOR EQUIVALENT CIRCUIT PARAMETERS (see Figure 16)

PARAMETER	OPTION	SYMBOL	CONDITION	MIN	TYP	MAX	UNIT
Transconductance	Osc.1	$g_m$	$T = +25^{\circ}\text{C}, V_{DD} = 4.5\text{V}$	—	15	—	$\mu\text{S}$
	Osc.2	$g_m$	$T = +25^{\circ}\text{C}, V_{DD} = 4.5\text{V}$	200	600	1000	$\mu\text{S}$
	Osc.3	$g_m$	$T = +25^{\circ}\text{C}, V_{DD} = 4.5\text{V}$	400	1500	4000	$\mu\text{S}$
	Osc.4	$g_m$	$T = +25^{\circ}\text{C}, V_{DD} = 4.5\text{V}$	1000	4000	10000	$\mu\text{S}$
Input capacitance	Osc.1	$C1_i$		—	3.0	—	pF
	Osc.2	$C1_i$		—	8.0	—	pF
	Osc.3	$C1_i$		—	8.0	—	pF
	Osc.4	$C1_i$		—	8.0	—	pF
Output capacitance	Osc.1	$C2_i$		—	23	—	pF
	Osc.2	$C2_i$		—	8.0	—	pF
	Osc.3	$C2_i$		—	8.0	—	pF
	Osc.4	$C2_i$		—	8.0	—	pF
Output resistance	Osc.1	$R2$		—	3800	—	$\text{k}\Omega$
	Osc.2	$R2$		—	65	—	$\text{k}\Omega$
	Osc.3	$R2$		—	18	—	$\text{k}\Omega$
	Osc.4	$R2$		—	5.0	—	$\text{k}\Omega$

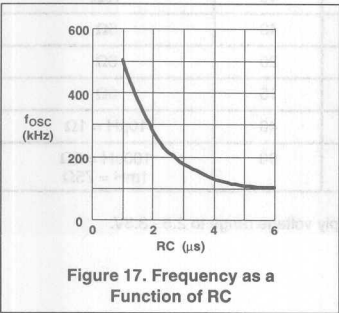
NOTE:

1. The equivalent circuit data of internal oscillator compares with that of matched crystals.



12.8.3 RC Oscillator

The externally adjustable RC-oscillator has a frequency range from 100kHz to 500kHz.





## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

### 12.9 Reset Circuitry

To initialize the 83CL781/83CL782, a reset is performed by either of two methods:

- via the RST pin
- via a power-on-reset.

It leaves the internal registers as follows:

Register	Content
ACC	0000 0000
B	0000 0000
DPL	0000 0000
DPH	0000 0000
IE0	0000 0000
IE1	0000 0000
IP0	XX00 0000
IP1	0000 0000
IX1	0000 0000
IRQ1	0000 0000
PCH	0000 0000
PCL	0000 0000
PCON	0XX0 0000
P0 - P3	1111 1111
S0BUF	XXXXXX
S0CON	0000 0000
S1ADR	0000 0000
S1CON	0000 0000
S1DAT	0000 0000
S1STA	1111 1000
SP	0000 0111
TCON	0000 0000
T2CON	0000 0000
T3	0000 0000
TH0, TH1, TH2	0000 0000
TL0, TL1, TL2	0000 0000
TMOD	0000 0000
PSW	0000 0000
RCAP2L	0000 0000
RCAP2H	0000 0000

The reset state of the port pins is mask-programmable and can therefore be defined by the user. The standard reset value for port P0-P3 is 1111 1111.

The reset input to the 83CL781/83CL782 is RST pin 15. A Schmitt trigger qualifies the input for noise rejection. The output of the Schmitt trigger is sampled by the reset circuitry every machine cycle.

A reset is accomplished by holding the RST pin HIGH for at least two machine cycles (24 oscillator periods), while the oscillator is running. The CPU responds by generating an internal reset. Port pins adopt their reset state immediately after RST goes HIGH. During reset ALE and PSEN are held HIGH.

The external reset is asynchronous to the internal clock. The RST pin is sampled during State 5, Phase 2 of every machine cycle. After a HIGH is detected at the RST pin, an internal reset is repeated every cycle until RST goes LOW.

The internal RAM is not affected by reset. When  $V_{DD}$  is turned on the RAM contents are indeterminate.

#### 1.9.1 Power-on reset

The 83CL781/83CL782 contains on-chip circuitry which switch the port pins to the

customer defined logic level as soon as  $V_{DD}$  exceeds 1.3V. As soon as the minimum supply voltage is reached, the oscillator will start up. However, to ensure that the oscillator is stable before the controller starts, the clock signals are gated away from the CPU for a further 1536 oscillator periods.

A hysteresis of approximately 50mV at a typical power-on switching level of 1.3V will ensure correct operation.

The on-chip Power-on reset circuitry can also be switched off via the mask option "OFF". This option reduces the Power-down current to typically 800nA and can be chosen if external reset circuitry is used.

An automatic reset can be obtained at power-on by connecting the RST pin to  $V_{DD}$  via a 10 $\mu$ F capacitor. At power-on, the voltage on the RST pin is equal to  $V_{DD}$  minus the capacitor voltage, and decreases from  $V_{DD}$  as the capacitor discharges through the internal resistor  $R_{RST}$  to ground. The larger the capacitor, the more slowly  $V_{RST}$  decreases.  $V_{RST}$  must remain above the lower threshold of the Schmitt trigger long enough to effect a complete reset. The time required is the oscillator start-up time, plus 2 machine cycles.

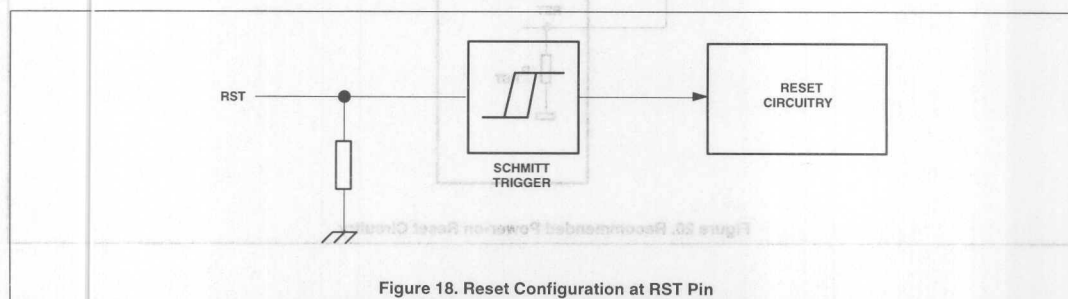
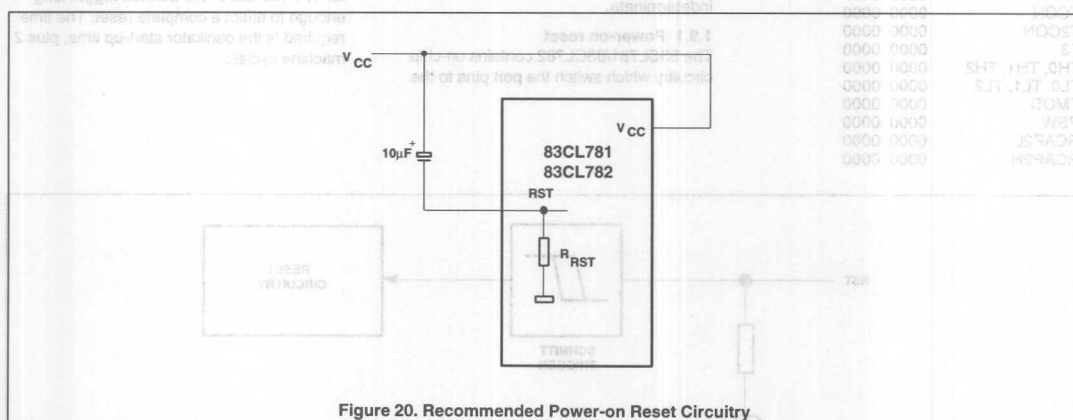
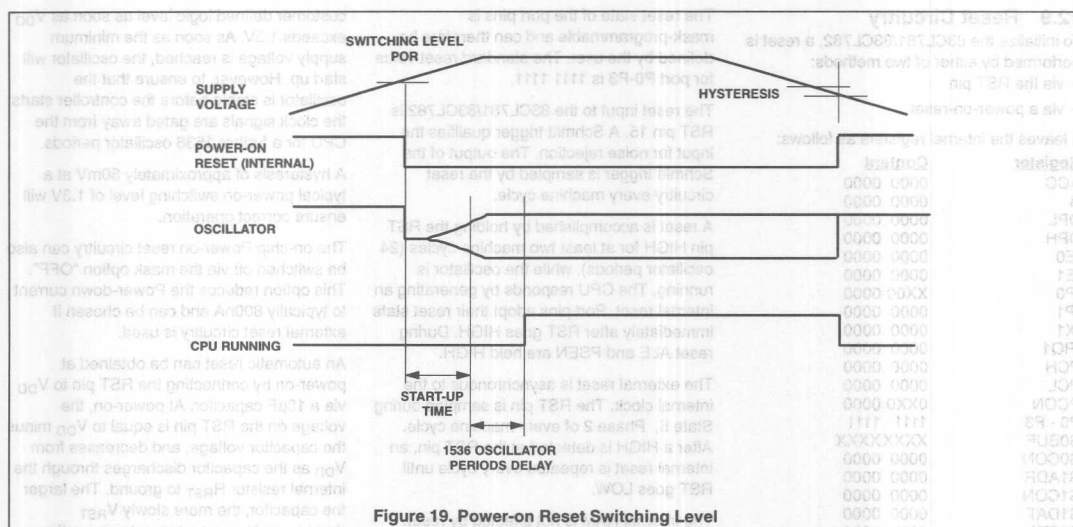


Figure 18. Reset Configuration at RST Pin

## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782



## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

## 13.0 RATINGS

## ABSOLUTE MAXIMUM RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

SYMBOL	PARAMETER	LIMITS		UNIT
		MIN	MAX	
$V_{DD}$	Supply voltage	-0.5	6.5	V
$V_I$	All input voltages	-0.5	$V_{DD}+0.5$	V
$I_I, I_O$	DC current into any input or output	—	5	mA
$P_{tot}$	Total power dissipation	—	300	mW
$T_{stg}$	Storage temperature range	-65	+150	°C
$T_{amb}$	Operating ambient temperature range	-40	+85	°C
$T_J$	Operating junction temperature	—	125	°C

$A_{int}$	3	0.75	—	$V_{DD} = 3V, f_{osc} = 3.58 \text{ MHz}$	
$A_{ext}$	10	—	—	$V_{DD} = 1.8V, T_{amb} = 25^\circ\text{C}$	Power-down mode (note 3)
Inputs (note 3)					
$V_{IL}$	V	0.3V <sub>DD</sub>	—	$V_{DD}$	Input voltage LOW
$V_{IH}$	V	—	0.7V <sub>DD</sub>	$V_{DD}$	Input voltage HIGH
$I_{IL}$	$\mu\text{A}$	—	—	$V_{EE} < V_I < V_{DD}$	Input leakage current (port F0, EA)
Outputs					
$I_{OL}$	$\text{mA}$	—	1.8	$V_{DD} = 5V, V_{OH} = 0.4V$	Output sink current LOW, except 82A, 82C
	$\text{mA}$	—	0.7	$V_{DD} = 2.5V, V_{OH} = 0.4V$	
	$\text{mA}$	—	3.0	$V_{DD} = 5V, V_{OH} = 0.4V$	Output sink current 82A, 82C
$I_{OH}$	$\text{mA}$	—	1.8	$V_{DD} = 5V, V_{OH} = V_{DD}-0.4V$	Output source current HIGH, push-out options only
	$\text{mA}$	—	0.7	$V_{DD} = 2.5V, V_{OH} = V_{DD}-0.4V$	
$I_{IL}$	$\mu\text{A}$	100	—	$V_{DD} = 5V, V_{IH} = 0.4V$	Input current logic 0
	$\mu\text{A}$	50	—	$V_{DD} = 2.5V, V_{IH} = 0.4V$	
$I_{IL}$	$\text{mA}$	1.0	—	$V_{DD} = 5V, V_{IH} = V_{DD}/2$	Input current logic 1, 1-to-0 transition
	$\mu\text{A}$	300	—	$V_{DD} = 2.5V, V_{IH} = V_{DD}/2$	
$R_{ST}$	$\Omega$	—	10	—	RST pull-down resistor

NOTES:

- The operating supply current is measured with all output pins disconnected; XTAL1 driven with  $f = 0$ ;  $V_{DD} = 5V$ ;  $V_{IH} = V_{DD}$ ; XTAL2 not connected; EA = RST = Port 0 =  $V_{DD}$ .
- The idle mode supply current is measured with all output pins disconnected; XTAL1 driven with  $f = 0$ ;  $V_{DD} = 5V$ ;  $V_{IH} = V_{DD}$ ; XTAL2 not connected; EA = Port 0 =  $V_{DD}$ .
- The power-down current is measured with all output pins disconnected; XTAL1 not connected; EA = Port 0 =  $V_{DD}$ ; RST =  $V_{DD}$ ; XTAL2 not connected; EA = Port 0 =  $V_{DD}$ .
- Capacitive loading on Port 0 and Port 2 may cause spurious noise pulses to be superimposed on the LOW level output voltage of ALE. Port 1 and Port 3 pins when these pins make a 1-to-0 transition during bus operations in the most adverse conditions (capacitive loading >100pF) the noise pulses on the ALE line may exceed 0.5V. In this event it may be needed to qualify ALE with a Schmitt trigger. It was not address latch with a Schmitt trigger input.
- Capacitive loading on Port 0 and Port 2 may cause the HIGH level output voltage on ALE and PSEN to momentarily fall below the 0.7V of  $V_{DD}$  specification when the address bus is stabilizing.
- The input threshold voltage of P1, 82PCL and P1, 782DA meet the P0 bus specification, so an input voltage below 0.3V<sub>DD</sub> will be recognized as a logic 0, while an input voltage above 0.7V<sub>DD</sub> will be recognized as a logic 1.
- Circuits with PSEN or PSEN option "ON" are tested at  $V_{DD}$  minimum = 1.8V.
- With option "ON" (typical) 1.3V are tested at  $V_{DD}$  minimum = 2.5V.

## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

## 14.0 DC ELECTRICAL CHARACTERISTICS

## DC ELECTRICAL CHARACTERISTICS (83CL781)

$V_{DD} = 1.8V$  to  $6V$ ;  $V_{SS} = 0V$ ;  $T_{amb} = -40^{\circ}C$  to  $+85^{\circ}C$ , all voltages with respect to  $V_{SS}$  unless otherwise specified.

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP	MAX	
Supply Voltage						
V <sub>DD</sub>	Supply voltage		1.8	—	6.0	V
	RAM retention voltage in power-down mode		1.0	—	6.0	V
Supply Current						
I <sub>DD</sub>	Operating (note 1)	V <sub>DD</sub> = 5V; f <sub>clk</sub> = 12 MHz	—	17	25	mA
		V <sub>DD</sub> = 3V; f <sub>clk</sub> = 3.58 MHz	—	2.4	5	mA
	Idle mode (note 2)	V <sub>DD</sub> = 5V; f <sub>clk</sub> = 12 MHz	—	5.1	12	mA
		V <sub>DD</sub> = 3V; f <sub>clk</sub> = 3.58 MHz	—	0.75	3	mA
I <sub>PD</sub>	Power-down mode (note 3)	V <sub>DD</sub> = 1.8V; T <sub>amb</sub> = 25°C	—	—	10	μA
Inputs (note 6)						
V <sub>IL</sub>	Input voltage LOW		V <sub>SS</sub>	—	0.3V <sub>DD</sub>	V
V <sub>IH</sub>	Input voltage HIGH		0.7V <sub>DD</sub>	—	V <sub>DD</sub>	V
±I <sub>LI</sub>	Input leakage current (port P0, E <sub>A</sub> )	V <sub>SS</sub> < V <sub>I</sub> < V <sub>DD</sub>	—	—	10	μA
Outputs						
I <sub>OL</sub>	Output sink current LOW, except SDA, SCL	V <sub>DD</sub> = 5V; V <sub>OL</sub> = 0.4V	1.6	—	—	mA
		V <sub>DD</sub> = 2.5V; V <sub>OL</sub> = 0.4V	0.7	—	—	mA
	Output sink current, SDA, SCL	V <sub>DD</sub> = 5V; V <sub>OL</sub> = 0.4V	3.0	—	—	mA
-I <sub>OH</sub>	Output source current HIGH, push-pull options only	V <sub>DD</sub> = 5V; V <sub>OH</sub> = V <sub>DD</sub> -0.4V	1.6	—	—	mA
		V <sub>DD</sub> = 2.5V; V <sub>OH</sub> = V <sub>DD</sub> -0.4V	0.7	—	—	mA
-I <sub>IL</sub>	Input current logic 0	V <sub>DD</sub> = 5V; V <sub>IN</sub> = 0.4V	—	—	100	μA
		V <sub>DD</sub> = 2.5V; V <sub>IN</sub> = 0.4V	—	—	50	μA
-I <sub>TL</sub>	Input current logic 0, 1-to-0 transition	V <sub>DD</sub> = 5V; V <sub>IN</sub> = V <sub>DD</sub> /2	—	—	1.0	mA
		V <sub>DD</sub> = 2.5V; V <sub>IN</sub> = V <sub>DD</sub> /2	—	—	500	μA
R <sub>RST</sub>	RST pull-down resistor		10	—	200	kΩ

## NOTES:

- The operating supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 10\text{ ns}$ ;  $V_{IL} = V_{SS}$ ;  $V_{IH} = V_{DD}$ ; XTAL2 not connected; EA = RST + Port 0 =  $V_{DD}$ .
- The idle mode supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 10\text{ ns}$ ;  $V_{IL} = V_{SS}$ ;  $V_{IH} = V_{DD}$ ; XTAL2 not connected; EA = Port 0 =  $V_{DD}$ .
- The power-down current is measured with all output pins disconnected; XTAL1 not connected; EA = Port 0 =  $V_{DD}$ ; RST =  $V_{SS}$ .
- Capacitive loading on Port 0 and Port 2 may cause spurious noise pulses to be superimposed on the LOW level output voltage of ALE, Port 1 and Port 3 pins when these pins make a 1-to-0 transition during bus operations. In the most adverse conditions (capacitive loading  $>100\text{ pF}$ ) the noise pulse on the ALE line may exceed  $0.8V$ . In this event it may be required to qualify ALE with a Schmitt trigger, or use an address latch with a Schmitt trigger strobe input.
- Capacitive loading on Port 0 and Port 2 may cause the HIGH level output voltage on ALE and PSEN to momentarily fall below the  $0.9V$  of  $V_{DD}$  specification when the address bits are stabilizing.
- The input threshold voltage of P1.6/SCL and P1.7/SDA meet the I<sup>2</sup>C-bus specification, so an input voltage below  $0.3V_{DD}$  will be recognized as a logic 0, while an input voltage above  $0.7V_{DD}$  will be recognized as a logic 1.
- Circuits with Power-on Reset option "Off" are tested at  $V_{DD}$  minimum =  $1.8V$ ; with option "On" (typically  $1.3V$ ) are tested at  $V_{DD}$  minimum =  $2.3V$ .

## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

## DC ELECTRICAL CHARACTERISTICS (83CL782)

$V_{DD} = 1.8V$  to  $6V$ ;  $V_{SS} = 0V$ ;  $T_{amb} = -25^{\circ}C$  to  $+55^{\circ}C$ , all voltages with respect to  $V_{SS}$ ; unless otherwise specified.

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			
			MIN	TYP	MAX	UNIT
Supply Voltage						
V <sub>DD</sub>	Supply voltage		1.8	—	6.0	V
	RAM retention voltage in power-down mode		1.0	—	6.0	V
Supply Current						
I <sub>DD</sub>	Operating (note 1)	V <sub>DD</sub> = 3.1V; f <sub>clk</sub> = 12MHz	—	8.4	12	mA
		V <sub>DD</sub> = 3V; f <sub>clk</sub> = 3.58MHz	—	2.4	5	mA
	Idle mode (note 2)	V <sub>DD</sub> = 3.1V; f <sub>clk</sub> = 12MHz	—	2.7	5	mA
		V <sub>DD</sub> = 3V; f <sub>clk</sub> = 3.58MHz	—	0.75	3	mA
I <sub>PD</sub>	Power-down mode (note 3)	V <sub>DD</sub> = 1.8V; T <sub>amb</sub> = 25°C	—	—	10	μA
Inputs (note 6)						
V <sub>IL</sub>	Input voltage LOW		V <sub>SS</sub>	—	0.3V <sub>DD</sub>	V
V <sub>IH</sub>	Input voltage HIGH		0.7V <sub>DD</sub>	—	V <sub>DD</sub>	V
±I <sub>LI</sub>	Input leakage current (port 0, EA)	V <sub>SS</sub> < V <sub>I</sub> < V <sub>DD</sub>	—	—	10	μA
Outputs						
I <sub>OL</sub>	Output sink current LOW, except SDA, SCL	V <sub>DD</sub> = 5V; V <sub>OL</sub> = 0.4V	1.6	—	—	mA
		V <sub>DD</sub> = 2.5V; V <sub>OL</sub> = 0.4V	0.7	—	—	mA
	Output sink current, SDA, SCL	V <sub>DD</sub> = 5V; V <sub>OL</sub> = 0.4V	3.0	—	—	mA
-I <sub>OH</sub>	Output source current HIGH, push-pull options only	V <sub>DD</sub> = 5V; V <sub>OH</sub> = V <sub>DD</sub> - 0.4V	1.6	—	—	mA
		V <sub>DD</sub> = 2.5V; V <sub>OH</sub> = V <sub>DD</sub> - 0.4V	0.7	—	—	mA
-I <sub>IL</sub>	Input current logic 0	V <sub>DD</sub> = 5V; V <sub>IN</sub> = 0.4V	—	—	100	μA
		V <sub>DD</sub> = 2.5V; V <sub>IN</sub> = 0.4V	—	—	50	μA
-I <sub>ITL</sub>	Input current logic 0, 1-to-0 transition	V <sub>DD</sub> = 5V; V <sub>IN</sub> = V <sub>DD</sub> /2	—	—	1.0	mA
		V <sub>DD</sub> = 2.5V; V <sub>IN</sub> = V <sub>DD</sub> /2	—	—	500	μA
R <sub>RST</sub>	RST pull-down resistor		10	—	200	kΩ

## NOTES:

- The operating supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 10ns$ ;  $V_{IL} = V_{SS}$ ;  $V_{IH} = V_{DD}$ ; XTAL2 not connected; EA = RST + Port 0 =  $V_{DD}$ .
- The idle mode supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 10ns$ ;  $V_{IL} = V_{SS}$ ;  $V_{IH} = V_{DD}$ ; XTAL2 not connected; EA = Port 0 =  $V_{DD}$ .
- The power-down current is measured with all output pins disconnected; XTAL1 not connected; EA = Port 0 =  $V_{DD}$ ; RST =  $V_{SS}$ .
- Capacitive loading on Port 0 and Port 2 may cause spurious noise pulses to be superimposed on the LOW level output voltage of ALE, Port 1 and Port 3 pins when these pins make a 1-to-0 transition during bus operations. In the most adverse conditions (capacitive loading >100 pF) the noise pulse on the ALE line may exceed 0.8V. In this event it may be required to qualify ALE with a Schmitt trigger, or use an address latch with a Schmitt trigger strobe input.
- Capacitive loading on Port 0 and Port 2 may cause the HIGH level output voltage on ALE and PSEN to momentarily fall below the 0.9% of  $V_{DD}$  specification when the address bits are stabilizing.
- The input threshold voltage of P1.6/SCL and P1.7/SDA meet the I<sup>2</sup>C-bus specification, so an input voltage below  $0.3V_{DD}$  will be recognized as a logic 0, while an input voltage above  $0.7V_{DD}$  will be recognized as a logic 1.
- Circuits with Power-on Reset option "Off" are tested at  $V_{DD}$  minimum = 1.8V; with option "On" (typically 1.3V), they are tested at  $V_{DD}$  minimum = 2.3V.



## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

### 15.0 AC CHARACTERISTICS

#### AC ELECTRICAL CHARACTERISTICS (83CL781)

$V_{DD} = 5V$ ;  $V_{SS} = 0V$ ;  $T_{amb} = -40^{\circ}C$  to  $+85^{\circ}C$ ;  $C_L = 50pF$  for Port 0, ALE and PSEN;  $C_L = 80pF$  for all other outputs unless otherwise specified.

SYMBOL	PARAMETER	12MHz CLOCK		VARIABLE CLOCK		UNIT
		MIN	MAX	MIN	MAX	
Program Memory (see Figure 21)						
t <sub>LL</sub>	ALE pulse duration	127	—	2t <sub>CK</sub> -40	—	ns
t <sub>AL</sub>	Address set-up time to ALE	43	—	t <sub>CK</sub> -40	—	ns
t <sub>LA</sub>	Address hold time after ALE	48	—	t <sub>CK</sub> -35	—	ns
t <sub>LC</sub>	Time from ALE to control pulse PSEN	58	—	t <sub>CK</sub> -25	—	ns
t <sub>LIV</sub>	Time from ALE to valid instruction input	—	233	—	4t <sub>CK</sub> -100	ns
t <sub>CC</sub>	Control pulse duration PSEN	215	—	3t <sub>CK</sub> -35	—	ns
t <sub>CIV</sub>	Time from PSEN to valid instruction input	—	125	—	3t <sub>CK</sub> -125	ns
t <sub>CI</sub>	Input instruction hold time after PSEN	0	—	0	—	ns
t <sub>CIF</sub>	Input instruction float delay after PSEN	—	63	—	t <sub>CK</sub> -20	ns
t <sub>AC</sub>	Address valid after PSEN	75	—	t <sub>CK</sub> -8	—	ns
t <sub>AIV</sub>	Address to valid instruction input	—	302	—	5t <sub>CK</sub> -115	ns
t <sub>AFC</sub>	Address float time to PSEN	12	—	0	—	ns

#### AC ELECTRICAL CHARACTERISTICS (83CL781)

$V_{DD} = 5V$ ;  $V_{SS} = 0V$ ;  $T_{amb} = -40^{\circ}C$  to  $+85^{\circ}C$ ;  $C_L = 50pF$  for Port 0; ALE and PSEN,  $C_L = 40pF$  for all other outputs unless otherwise specified.

SYMBOL	PARAMETER	12MHz CLOCK		VARIABLE CLOCK		UNIT
		MIN	MAX	MIN	MAX	
External Data Memory (see Figures 22 and 23)						
t <sub>RR</sub>	RD pulse duration	400	—	6t <sub>CK</sub> -100	—	ns
t <sub>WW</sub>	WR pulse duration	400	—	6t <sub>CK</sub> -100	—	ns
t <sub>LA</sub>	Address hold time after ALE	48	—	t <sub>CK</sub> -35	—	ns
t <sub>RD</sub>	RD to valid data input	—	150	—	5t <sub>CK</sub> -165	ns
t <sub>DFR</sub>	Data float delay after RD	—	97	—	2t <sub>CK</sub> -70	ns
t <sub>LD</sub>	Time from ALE to valid data input	—	517	—	8t <sub>CK</sub> -150	ns
t <sub>AD</sub>	Address to valid data input	—	585	—	9t <sub>CK</sub> -165	ns
t <sub>LW</sub>	Time from ALE to RD and WR	200	300	3t <sub>CK</sub> -50	3t <sub>CK</sub> +50	ns
t <sub>AW</sub>	Time from address to RD or WR	203	—	4	—	ns
t <sub>WHLH</sub>	Time from RD or WR HIGH to ALE HIGH	43	123	t <sub>CK</sub> -130 t <sub>CK</sub> -40	t <sub>CK</sub> +40	ns
t <sub>DWX</sub>	Data valid to WR transition	23	—	t <sub>CK</sub> -60	—	ns
t <sub>DW</sub>	Data set-up time before WR	433	—	7t <sub>CK</sub> -150	—	ns
t <sub>WD</sub>	Data hold time after WR	33	—	t <sub>CK</sub> -50	—	ns
t <sub>AFR</sub>	Address float delay after RD	—	12	—	12	ns

#### NOTE TO THE AC CHARACTERISTICS:

- Interfacing the 83CL781 to devices with float times up to 75ns is permitted. This limited bus contention will not cause damage to Port 0 drivers.

## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

**AC ELECTRICAL CHARACTERISTICS (83CL782)**V<sub>DD</sub> = 5V; V<sub>SS</sub> = 0V; T<sub>amb</sub> = -25°C to +55°C; C<sub>L</sub> = 50pF for Port 0, ALE and PSEN; C<sub>L</sub> = 80pF for all other outputs unless otherwise specified.

SYMBOL	PARAMETER	12MHz CLOCK		VARIABLE CLOCK		UNIT
		MIN	MAX	MIN	MAX	
Program Memory (see Figure 21)						
f <sub>CL</sub>	Clock frequency (V <sub>DD</sub> = 2.7V)	32kHz	12MHz	—	—	
t <sub>LL</sub>	ALE pulse duration	127	—	2t <sub>CK</sub> -40	—	ns
t <sub>AL</sub>	Address set-up time to ALE	43	—	t <sub>CK</sub> -40	—	ns
t <sub>LA</sub>	Address hold time after ALE	48	—	t <sub>CK</sub> -35	—	ns
t <sub>LC</sub>	Time from ALE to control pulse PSEN	58	—	t <sub>CK</sub> -25	—	ns
t <sub>LIV</sub>	Time from ALE to valid instruction input	—	233	—	4t <sub>CK</sub> -100	ns
t <sub>CC</sub>	Control pulse duration PSEN	215	—	3t <sub>CK</sub> -35	—	ns
t <sub>CIv</sub>	Time from PSEN to valid instruction input	—	125	—	3t <sub>CK</sub> -125	ns
t <sub>CI</sub>	Input instruction hold time after PSEN	0	—	0	—	ns
t <sub>CIv</sub>	Input instruction float delay after PSEN	—	63	—	t <sub>CK</sub> -20	ns
t <sub>AC</sub>	Address valid after PSEN	75	—	t <sub>CK</sub> -8	—	ns
t <sub>AIV</sub>	Address to valid instruction input	—	302	—	5t <sub>CK</sub> -115	ns
t <sub>AFC</sub>	Address float time to PSEN	12	—	0	—	ns

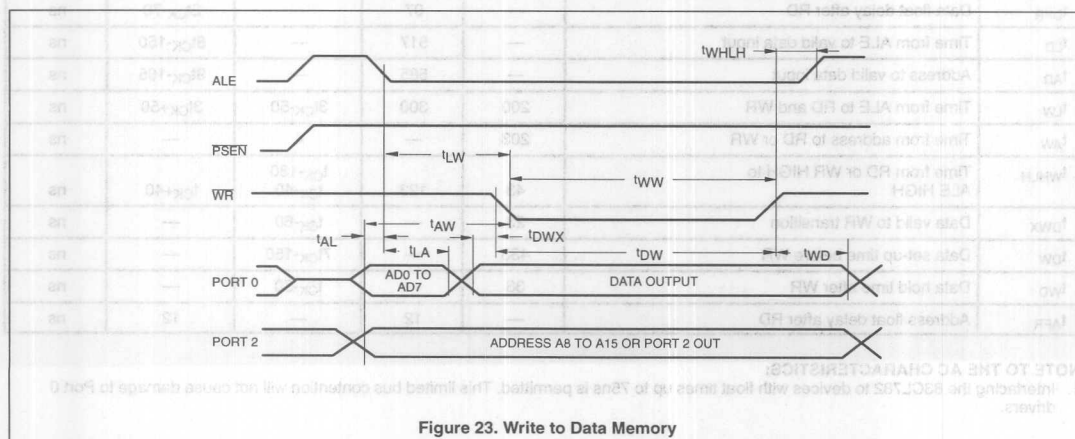
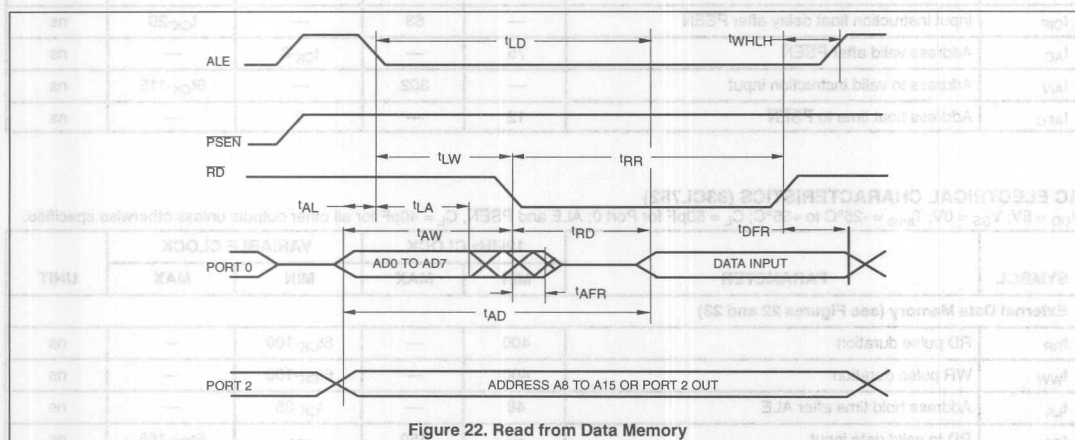
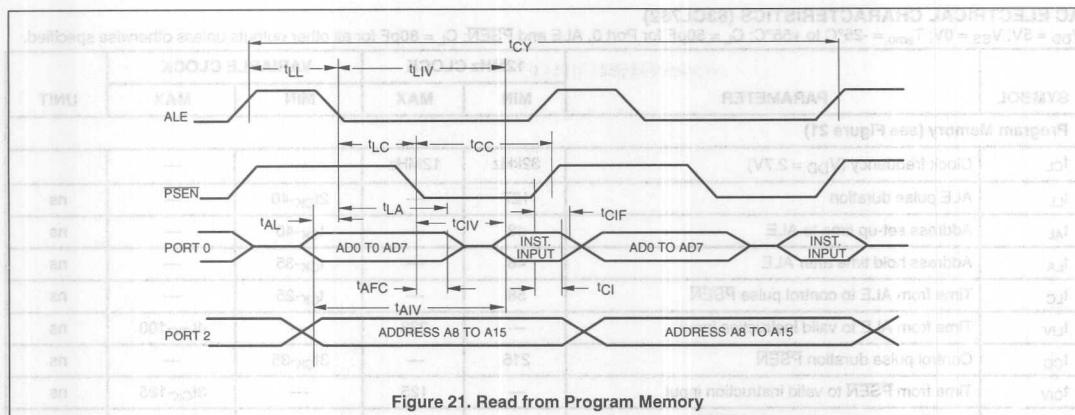
**AC ELECTRICAL CHARACTERISTICS (83CL782)**V<sub>DD</sub> = 5V; V<sub>SS</sub> = 0V; T<sub>amb</sub> = -25°C to +55°C; C<sub>L</sub> = 50pF for Port 0; ALE and PSEN, C<sub>L</sub> = 40pF for all other outputs unless otherwise specified.

SYMBOL	PARAMETER	12MHz CLOCK		VARIABLE CLOCK		UNIT
		MIN	MAX	MIN	MAX	
External Data Memory (see Figures 22 and 23)						
t <sub>RR</sub>	RD pulse duration	400	—	6t <sub>CK</sub> -100	—	ns
t <sub>WW</sub>	WR pulse duration	400	—	6t <sub>CK</sub> -100	—	ns
t <sub>LA</sub>	Address hold time after ALE	48	—	t <sub>CK</sub> -35	—	ns
t <sub>RD</sub>	RD to valid data input	—	150	—	5t <sub>CK</sub> -165	ns
t <sub>DFR</sub>	Data float delay after RD	—	97	—	2t <sub>CK</sub> -70	ns
t <sub>LD</sub>	Time from ALE to valid data input	—	517	—	8t <sub>CK</sub> -150	ns
t <sub>AD</sub>	Address to valid data input	—	585	—	9t <sub>CK</sub> -165	ns
t <sub>LW</sub>	Time from ALE to RD and WR	200	300	3t <sub>CK</sub> -50	3t <sub>CK</sub> +50	ns
t <sub>AW</sub>	Time from address to RD or WR	203	—	4	—	ns
t <sub>WHLH</sub>	Time from RD or WR HIGH to ALE HIGH	43	123	t <sub>CK</sub> -130 t <sub>CK</sub> -40	t <sub>CK</sub> +40	ns
t <sub>DWX</sub>	Data valid to WR transition	23	—	t <sub>CK</sub> -60	—	ns
t <sub>DW</sub>	Data set-up time before WR	433	—	7t <sub>CK</sub> -150	—	ns
t <sub>WD</sub>	Data hold time after WR	33	—	t <sub>CK</sub> -50	—	ns
t <sub>AFR</sub>	Address float delay after RD	—	12	—	12	ns

**NOTE TO THE AC CHARACTERISTICS:**

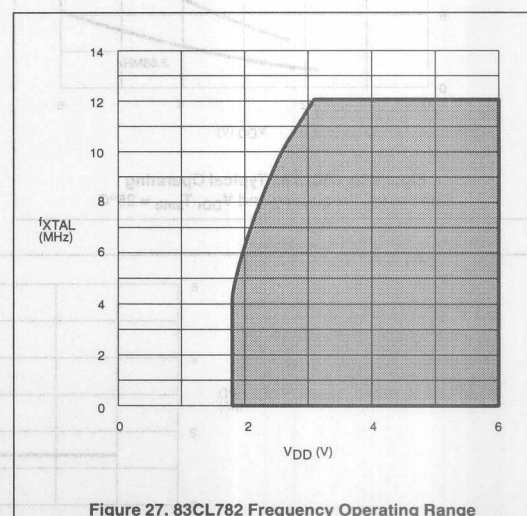
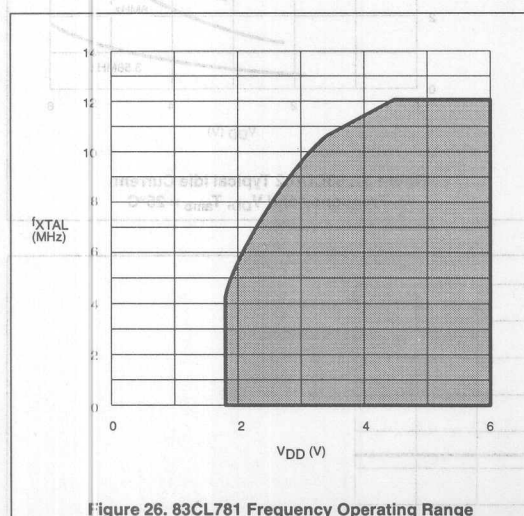
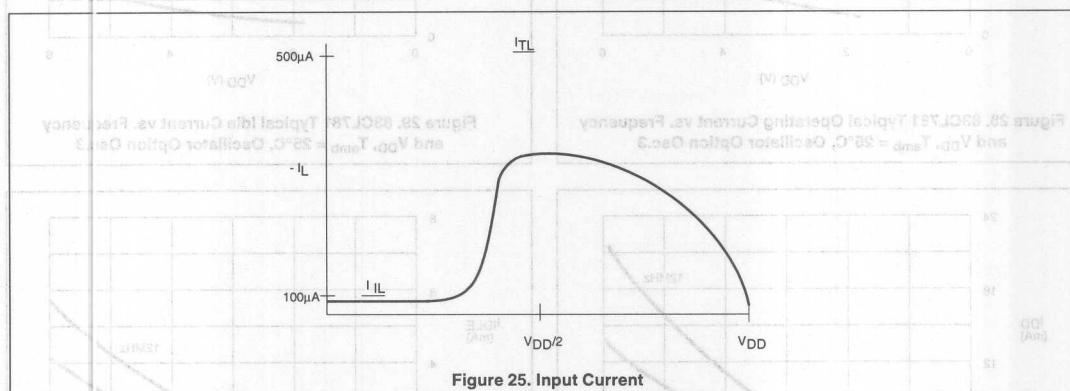
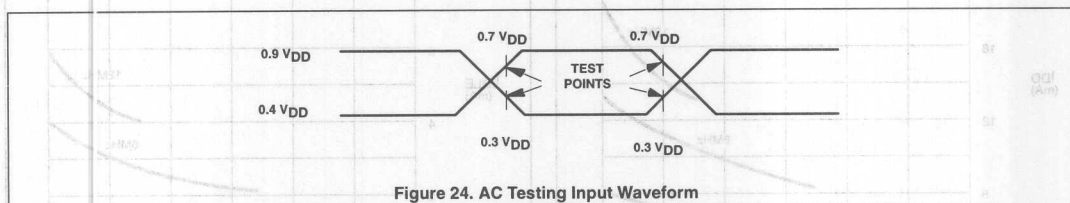
- Interfacing the 83CL782 to devices with float times up to 75ns is permitted. This limited bus contention will not cause damage to Port 0 drivers.

## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782



# Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

## 16.0 CHARACTERISTIC CURVES



## Low-voltage single-chip 8-bit microcontrollers 83CL781/83CL782

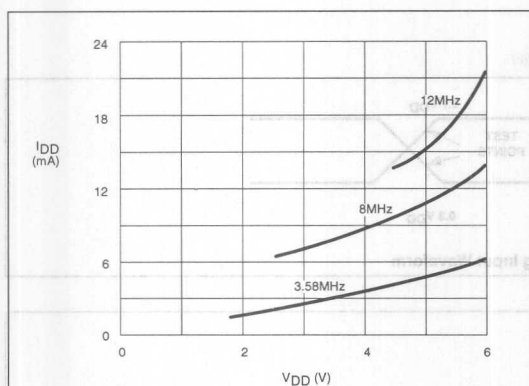


Figure 28. 83CL781 Typical Operating Current vs. Frequency and  $V_{DD}$ ,  $T_{amb} = 25^{\circ}\text{C}$ , Oscillator Option Osc.3

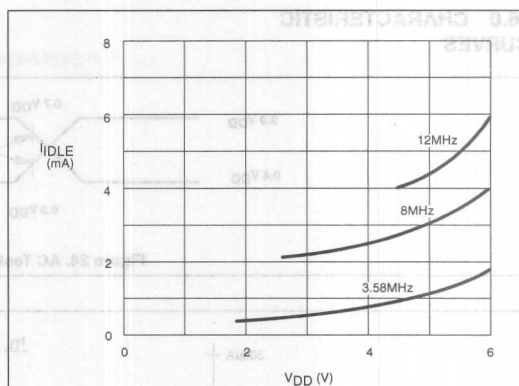


Figure 29. 83CL781 Typical Idle Current vs. Frequency and  $V_{DD}$ ,  $T_{amb} = 25^{\circ}\text{C}$ , Oscillator Option Osc.3

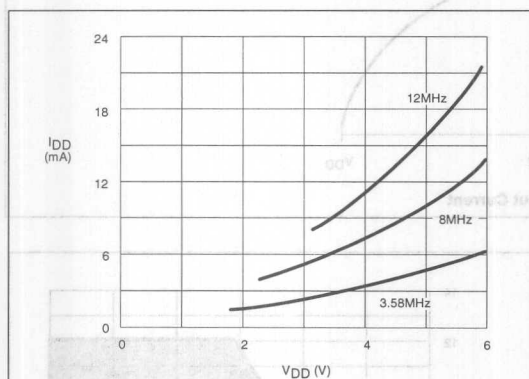


Figure 30. 83CL782 Typical Operating Current vs. Frequency and  $V_{DD}$ ,  $T_{amb} = 25^{\circ}\text{C}$

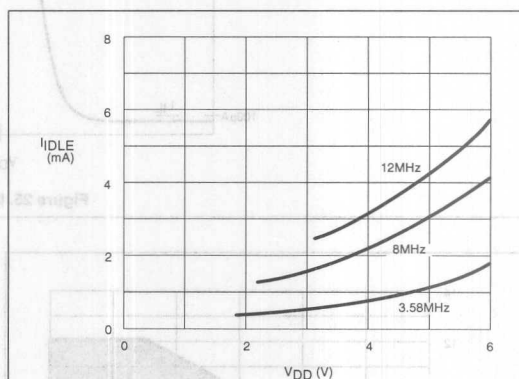


Figure 31. 83CL782 Typical Idle Current vs. Frequency and  $V_{DD}$ ,  $T_{amb} = 25^{\circ}\text{C}$

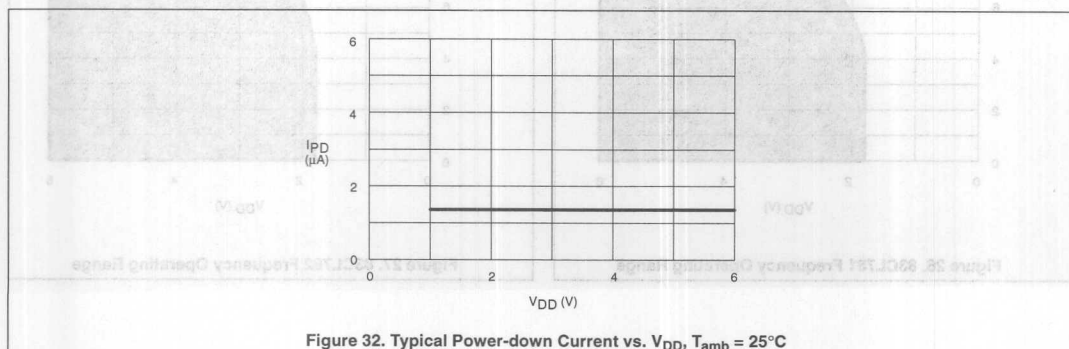


Figure 32. Typical Power-down Current vs.  $V_{DD}$ ,  $T_{amb} = 25^{\circ}\text{C}$



# CMOS single-chip 8-bit microcontroller with on-chip EEPROM

## 80C851/83C851

### DESCRIPTION

The Philips 80C851/83C851 is a high-performance microcontroller fabricated with Philips high-density CMOS technology. The 80C851/83C851 has the same instruction set as the 80C51. The Philips CMOS technology combines the high speed and density characteristics of HMOS with the low power attributes of CMOS. The Philips epitaxial substrate minimizes latch-up sensitivity.

The 80C851/83C851 contains a  $4k \times 8$  ROM with mask-programmable ROM code protection, a  $128 \times 8$  RAM,  $256 \times 8$  EEPROM, 32 I/O lines, two 16-bit counter/timers, a seven-source, five vector, two-priority level nested interrupt structure, a serial I/O port for either multi-processor communications, I/O expansion or full duplex UART, and on-chip oscillator and clock circuits.

In addition, the 80C851/83C851 has two software selectable modes of power reduction — idle mode and power-down mode. The idle mode freezes the CPU while allowing the RAM, timers, serial port, and interrupt system to continue functioning. The power-down mode saves the RAM and EEPROM contents but freezes the oscillator, causing all other chip functions to be inoperative.

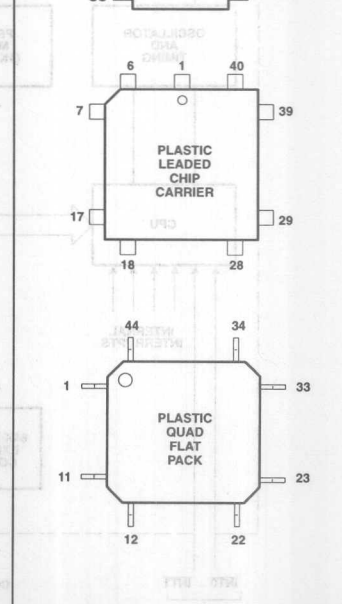
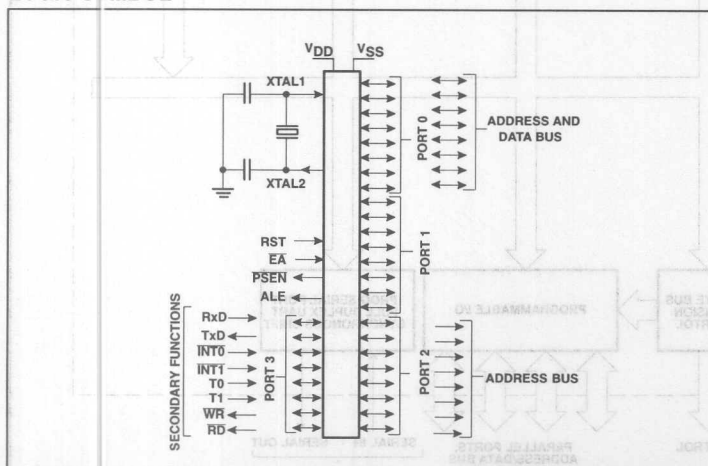
### FEATURES

- 80C51 based architecture
  - $4k \times 8$  ROM
  - $128 \times 8$  RAM
  - Two 16-bit counter/timers
  - Full duplex serial channel
  - Boolean processor
- Non-volatile  $256 \times 8$ -bit EEPROM (electrically erasable programmable read only memory)
  - On-chip voltage multiplier for erase/write
  - 50,000 erase/write cycles per byte
  - 10 years non-volatile data retention
  - Infinite number of read cycles
  - User selectable security mode
  - Block erase capability
- Mask-programmable ROM code protection
- Memory addressing capability
  - $64k$  ROM and  $64k$  RAM
- Power control modes:
  - Idle mode
  - Power-down mode
- CMOS and TTL compatible
- 1.2 to 16MHz
- Three package styles
- Three temperature ranges
- ROM code protection

### PIN CONFIGURATIONS

P1.0	1	40	VDD
P1.1	2	39	P0.0/AD0
P1.2	3	38	P0.1/AD1
P1.3	4	37	P0.2/AD2
P1.4	5	36	P0.3/AD3
P1.5	6	35	P0.4/AD4
P1.6	7	34	P0.5/AD5
P1.7	8	33	P0.6/AD6
RST	9	32	P0.7/AD7
RxD/P3.0	10	31	EA
TxD/P3.1	11	30	ALE
INT0/P3.2	12	29	PSEN
INT1/P3.3	13	28	P2.7/A15
T0/P3.4	14	27	P2.6/A14
T1/P3.5	15	26	P2.5/A13
WR/P3.6	16	25	P2.4/A12
RD/P3.7	17	24	P2.3/A11
XTAL2	18	23	P2.2/A10
XTAL1	19	22	P2.1/A9
VSS	20	21	P2.0/A8

### LOGIC SYMBOL



# CMOS single-chip 8-bit microcontroller with on-chip EEPROM

80C851/83C851

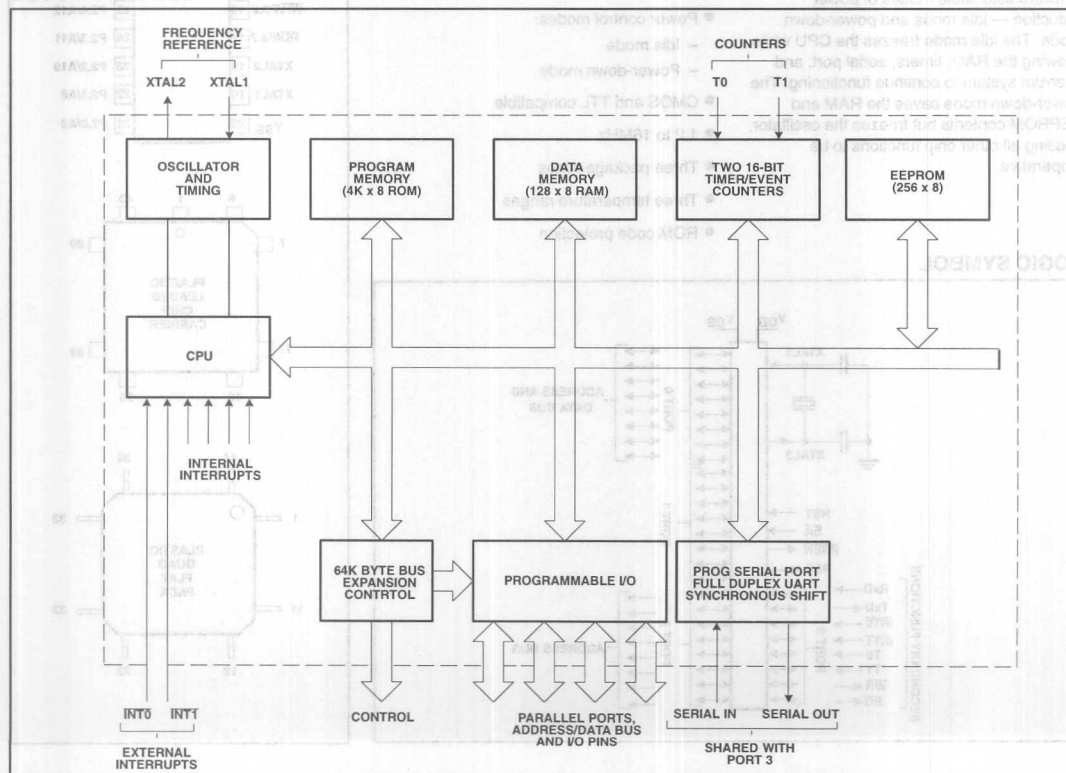
## ORDERING INFORMATION

PHILIPS PART ORDER NUMBER PART MARKING		NORTH AMERICA PHILIPS PART ORDER NUMBER		TEMPERATURE RANGE °C AND PACKAGE	FREQ. (MHz)	DRAWING NUMBER
ROMless Version	ROM Version	ROMless Version	ROM Version			
P80C851 FBP	P83C851 FBP	S80C851-4N40	S83C851-4N40	0 to +70, Plastic Dual In-line Package	1.2 to 16	SOT129
P80C851 FBA	P83C851 FBA	S80C851-4A44	S83C851-4A44	0 to +70, Plastic Leaded Chip Carrier	1.2 to 16	SOT187
P80C851 FBB	P83C851 FBB	S80C851-4B44	S83C851-4B44	0 to +70, Plastic Quad Flat Pack	1.2 to 16	SOT307-2 <sup>1</sup>
P80C851 FFP	P83C851 FFP	S80C851-5N40	S83C851-5N40	-40 to +85, Plastic Dual In-line Package	1.2 to 16	SOT129
P80C851 FFA	P83C851 FFA	S80C851-5A44	S83C851-5A44	-40 to +85, Plastic Leaded Chip Carrier	1.2 to 16	SOT187
P80C851 FFB	P83C851 FFB	S80C851-5B44	S83C851-5B44	-40 to +85, Plastic Quad Flat Pack	1.2 to 16	SOT307-2 <sup>1</sup>
P80C851 FHP	P83C851 FHP	S80C851-6N40	S83C851-6N40	-40 to +125, Plastic Dual In-line Package	1.2 to 16	SOT129
P80C851 FHA	P83C851 FHA	S80C851-6A44	S83C851-6A44	-40 to +125, Plastic Leaded Chip Carrier	1.2 to 16	SOT187
P80C851 FHB	P83C851 FHB	S80C851-6B44	S83C851-6B44	-40 to +125, Plastic Quad Flat Pack	1.2 to 16	SOT307-2 <sup>1</sup>

## NOTE:

1. SOT311 replaced by SOT307-2.

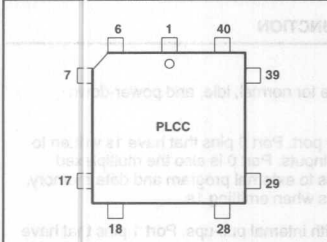
## BLOCK DIAGRAM



# CMOS single-chip 8-bit microcontroller with on-chip EEPROM

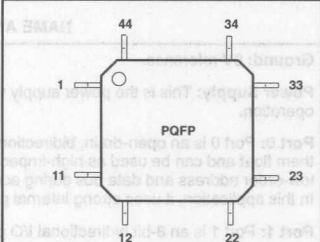
80C851/83C851

## PLASTIC LEADED CHIP CARRIER PIN FUNCTIONS



Pin	Function	Pin	Function
1	NC	23	NC
2	P1.0	24	P2.0/A8
3	P1.1	25	P2.1/A9
4	P1.2	26	P2.2/A10
5	P1.3	27	P2.3/A11
6	P1.4	28	P2.4/A12
7	P1.5	29	P2.5/A13
8	P1.6	30	P2.6/A14
9	P1.7	31	P2.7/A15
10	RST	32	PSEN
11	P3.0/RxD	33	ALE
12	NC	34	NC
13	P3.1/TxD	35	EA
14	P3.2/INT0	36	P0.7/AD7
15	P3.3/INT1	37	P0.6/AD6
16	P3.4/T0	38	P0.5/AD5
17	P3.5/T1	39	P0.4/AD4
18	P3.6/WR	40	P0.3/AD3
19	P3.7/RD	41	P0.2/AD2
20	XTAL2	42	P0.1/AD1
21	XTAL1	43	P0.0/AD0
22	VSS	44	VDD

## PLASTIC QUAD FLAT PACK PIN FUNCTIONS



Pin	Function	Pin	Function
1	NC	23	P2.5/A13
2	P1.6	24	P2.6/A14
3	P1.7	25	P2.7/A15
4	RST	26	PSEN
5	P3.0/RxD	27	ALE
6	NC	28	NC
7	P3.1/TxD	29	EA
8	P3.2/INT0	30	P0.7/AD7
9	P3.3/INT1	31	P0.6/AD6
10	P3.4/T0	32	P0.5/AD5
11	P3.5/T1	33	P0.4/AD4
12	P3.6/WR	34	P0.3/AD3
13	P3.7/RD	35	P0.2/AD2
14	XTAL2	36	P0.1/AD1
15	XTAL1	37	P0.0/AD0
16	VSS	38	VDD
17	NC	39	VSS
18	P2.0/A8	40	P1.0
19	P2.1/A9	41	P1.1
20	P2.2/A10	42	P1.2
21	P2.3/A11	43	P1.3
22	P2.4/A12	44	P1.4

# CMOS single-chip 8-bit microcontroller with on-chip EEPROM

80C851/83C851

## PIN DESCRIPTION

MNEMONIC	PIN NO.			TYPE	NAME AND FUNCTION
	DIP	LCC	QFP		
V <sub>SS</sub>	20	22	16, 39	I	<b>Ground:</b> 0V reference.
V <sub>DD</sub>	40	44	38	I	<b>Power Supply:</b> This is the power supply voltage for normal, idle, and power-down operation.
P0.0–0.7	39–32	43–36	37–30	I/O	<b>Port 0:</b> Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s.
P1.0–P1.7	1–8	2–9	40–44, 1–3	I/O	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ).
P2.0–P2.7	21–28	24–31	18–25	I/O	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register.
P3.0–P3.7	10–17	11, 13–19	5, 7–13	I/O	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 3 also serves the special features of the SC80C51 family, as listed below:
				I	<b>RxD (P3.0):</b> Serial input port
				O	<b>TxD (P3.1):</b> Serial output port
				I	<b>INT0 (P3.2):</b> External interrupt
				I	<b>INT1 (P3.3):</b> External interrupt
				I	<b>T0 (P3.4):</b> Timer 0 external input
				I	<b>T1 (P3.5):</b> Timer 1 external input
				O	<b>WR (P3.6):</b> External data memory write strobe
				O	<b>RD (P3.7):</b> External data memory read strobe
RST	9	10	4	I	<b>Reset:</b> A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to V <sub>SS</sub> permits a power-on reset using only an external capacitor to V <sub>DD</sub> .
ALE	30	33	27	I/O	<b>Address Latch Enable:</b> Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory.
PSEN	29	32	26	O	<b>Program Store Enable:</b> The read strobe to external program memory. When the device is executing code from the external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory. PSEN is not activated during fetches from internal program memory.
EA	31	35	29	I	<b>External Access Enable:</b> If during a RESET, EA is held at TTL level HIGH, the CPU executes out of the internal program memory ROM provided the Program Counter is less than 4096. If during a RESET, EA is held a TTL LOW level, the CPU executes out of external program memory. EA is not allowed to float.
XTAL1	19	21	15	I	<b>Crystal 1:</b> Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	20	14	O	<b>Crystal 2:</b> Output from the inverting oscillator amplifier.

# CMOS single-chip 8-bit microcontroller with on-chip EEPROM

80C851/83C851

Table 1. 8XC851 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB							LSB	
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
DPTR:	Data pointer (2 bytes):		EF	EE	ED	EC	EB	EA	E9	E8	
DPH	High byte	83H									00H
DPL	Low byte	82H									00H
EADRH#	EEPROM addr reg-high	F3H									80H
EADRL#	EEPROM addr reg-low	F2H									00H
ECNTRL#	EEPROM control reg	F6H	IFE	EEINT	EWP	ECNTR L3	ECNTR L2	ECNTR L1	ECNTR L0		00H
EDAT#	EEPROM data register	F4H									xxH
ETIM#	EEPROM timer register	F5H									08H
IP*	Interrupt priority	B8H	BF	BE	BD	BC	BB	BA	B9	B8	
			–	–	–	PS	PT1	PX1	PT0	PX0	xxx00000B
IE*	Interrupt enable	A8H	AF	AE	AD	AC	AB	AA	A9	A8	
			EA	–	–	ES	ET1	EX1	ET0	EX0	0xx00000B
P0*	Port 0	80H	87	86	85	84	83	82	81	80	FFH
P1*	Port 1	90H	97	96	95	94	93	92	91	90	FFH
P2*	Port 2	A0H	A7	A6	A5	A4	A3	A2	A1	A0	FFH
P3*	Port 3	B0H	B7	B6	B5	B4	B3	B2	B1	B0	FFH
PCON	Power control	87H	SMOD	–	–	–	GF1	GF0	PD	IDL	0xxx0000B
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program status word	D0H	CY	AC	F0	RS1	RS0	OV	–	P	00H
SBUF	Serial data buffer	99H									xxxxxxxxB
			9F	9E	9D	9C	9B	9A	99	98	
SCON*	Serial port control	98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00H
SP	Stack pointer	81H									07H
			8F	8E	8D	8C	8B	8A	89	88	00H
TCON*	Timer/counter control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
TMOD	Timer/counter mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H
TH0	Timer 0 high byte	8CH									00H
TH1	Timer 1 high byte	8DH									00H
TL0	Timer 0 low byte	8AH									00H
TL1	Timer 1 low byte	8BH									00H

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.



# CMOS single-chip 8-bit microcontroller with on-chip EEPROM

80C851/83C851

## EEPROM

Communications between the CPU and the EEPROM is accomplished via 5 special function registers; 2 address registers (high and low byte), 1 data register for read and write operations, 1 control register, and 1 timer register to adapt the erase/write time to the clock frequency. All registers can be read and written. Figure 1 shows a block diagram of the CPU, the EEPROM and the interface.

## Register and Functional Description

### Address Register (EADRH, EADRL)

The lower byte contains the address of one of the 256 bytes. The higher byte (EADRH) is for future extensions and for addressing the security bits (see Security Facilities). The

EADRH register address is F3H. The EADRL register address is F2H.

### Data Register (EDAT)

This register is required for read and write operations and also for row/block erase. In write mode, its contents are written to the addressed byte (for "row erase" and "block erase" the contents are don't care). The write pulse starts all operations, except read. In read mode, EDAT contains the data of the addressed byte. The EDAT register address is F4H.

### Timer Register (ETIM)

The timer register is required to adapt the erase/write time to the oscillator frequency. The user has to ensure that the erase or write (program) time is neither too short or too long.

The ETIM register address is F5H. Table 2 contains the values which must be written to the ETIM register by software for various oscillator frequencies (the default value is 08H after RESET).

The general formula is:

2ms Write time:

$$\text{Value (decimal, to be rounded up)} = \frac{f_{\text{XTAL1}} [\text{kHz}]}{512} - 2$$

10ms Write time:

$$\text{Value (decimal)} = \frac{f_{\text{XTAL1}} [\text{kHz}]}{96} - 2$$

### Control Register (ECNTRL)

See Figure 2 for a description of this register. The ECNTRL register address is F6H.

Table 2. Values for the Timer Register (ETIM)

f <sub>XTAL1</sub>	VALUES FOR ETIM			
	2ms WRITE TIME		10ms WRITE TIME	
	HEX	DEC	HEX	DEC
1.0MHz	—	—	08	8
2.0MHz	02	2	13	19
3.0MHz	04	4	1D	29
4.0MHz	06	6	28	40
5.0MHz	08	8	32	50
6.0MHz	0A	10	3C	60
7.0MHz	0C	12	47	71
8.0MHz	0E	14	51	81
9.0MHz	10	16	5C	92
10.0MHz	12	18	66	102
11.0MHz	14	20	71	113
12.0MHz	16	22	7B	123
13.0MHz	18	24		
14.0MHz	1A	26		
15.0MHz	1C	28		
16.0MHz	1E	30		

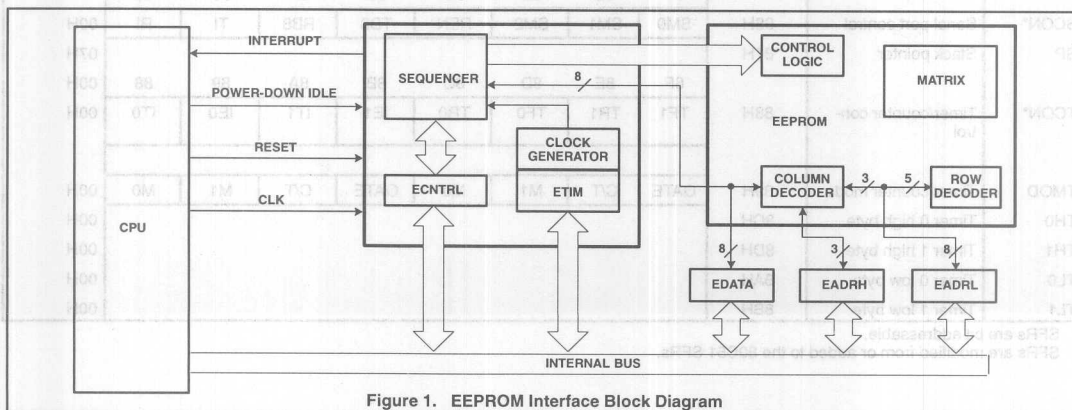
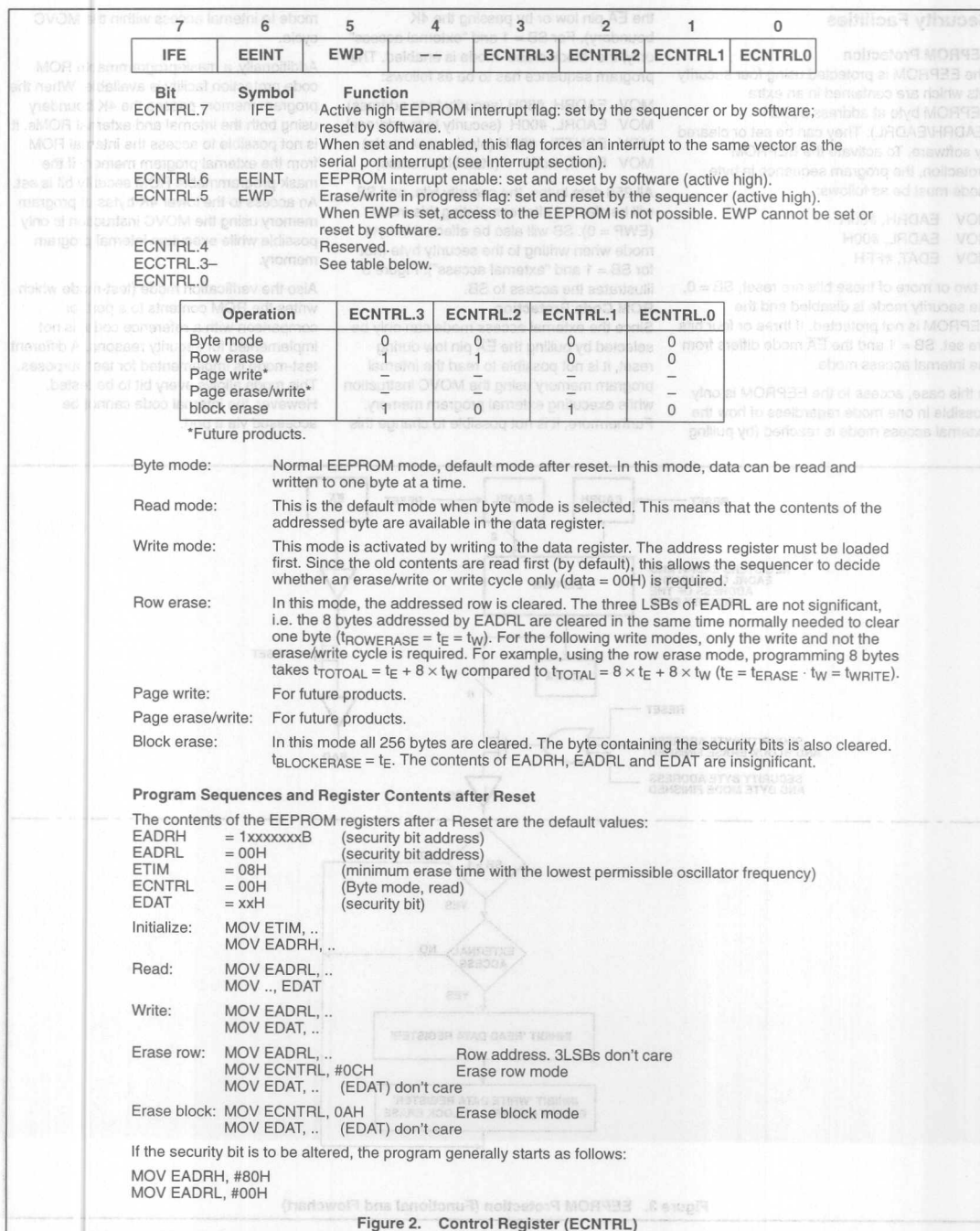


Figure 1. EEPROM Interface Block Diagram

# CMOS single-chip 8-bit microcontroller with on-chip EEPROM

80C851/83C851



## CMOS single-chip 8-bit microcontroller with on-chip EEPROM

80C851/83C851

## Security Facilities

### EEPROM Protection

The EEPROM is protected using four security bits which are contained in an extra EEPROM byte at address 8000H (EADRH/EADRL). They can be set or cleared by software. To activate the EEPROM protection, the program sequence in byte mode must be as follows:

```
MOV    EADRH, #80H
MOV    EADRL, #00H
MOV    EDAT, #FFH
```

If two or more of these bits are reset,  $SB = 0$ , the security mode is disabled and the EEPROM is not protected. If three or four bits are set,  $SB = 1$  and the  $\overline{EA}$  mode differs from the internal access mode.

In this case, access to the EEPROM is only possible in one mode regardless of how the external access mode is reached (by pulling

the EA pin low or by passing the 4K boundary). For SB = 1 and "external access" only, the "block erase" mode is enabled. The program sequence has to be as follows:

```
MOV EADRH, #80H (security byte address)
MOV EADRL, #00H (security byte address)
MOV ECNTRL, #0AH (block erase mode)
MOV EDAT, #xxH (start block erase)
```

All 256 data bytes, the security bits, and SB will be cleared after completing this mode (EWP = 0). SB will also be affected in byte mode when writing to the security byte (not for SB = 1 and "external access"). Figure 3 illustrates the access to SB.

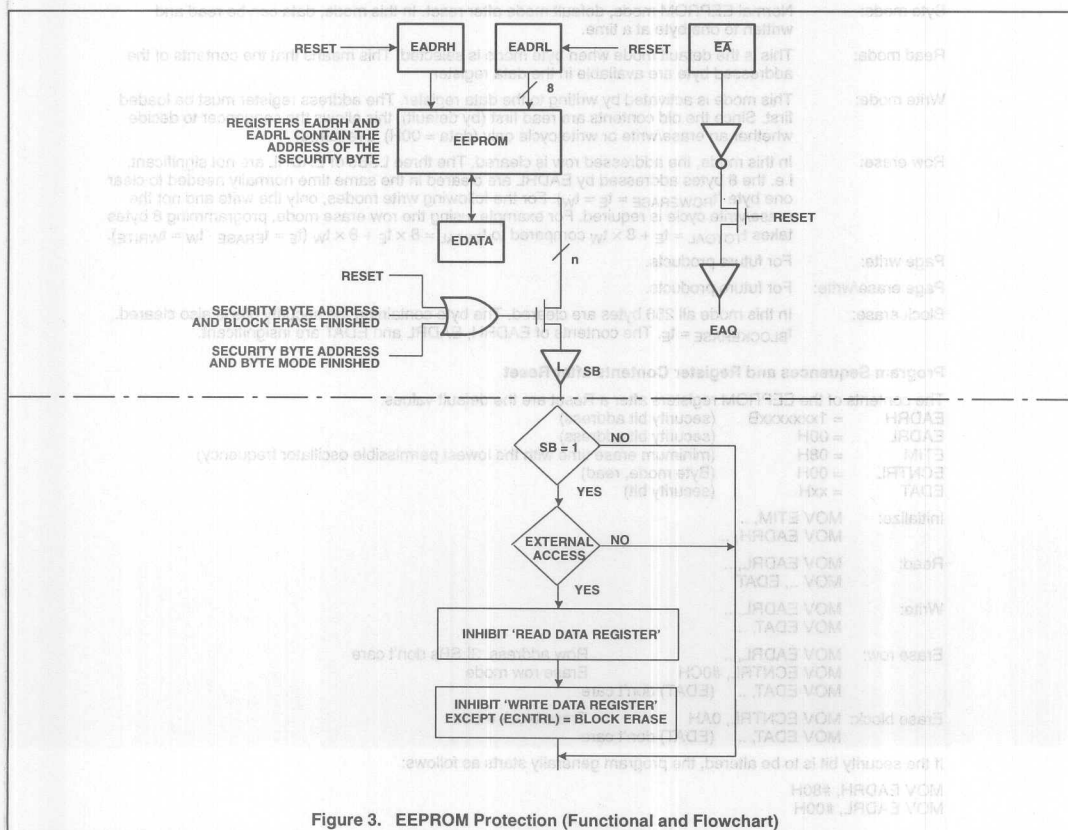
### ROM Code Protection

Since the external access mode can only be selected by pulling the  $\overline{\text{EA}}$  pin low during reset, it is not possible to read the internal program memory using the MOV<sub>C</sub> instruction while executing external program memory. Furthermore, it is not possible to change this

mode to internal access within the MOVc cycle.

Additionally, a mask-programmable ROM code protection facility is available. When the program memory passes the 4K boundary using both the internal and external ROMs, it is not possible to access the internal ROM from the external program memory if the mask-programmable ROM security bit is set. An access to the lower 4K bytes of program memory using the MOV<sub>C</sub> instruction is only possible while executing internal program memory.

Also the verification mode (test-mode which writes the ROM contents to a port for comparison with a reference code) is not implemented for security reasons. A different test-mode is implemented for test purposes. This mode allows every bit to be tested. However, the internal code cannot be accessed via a port.



# CMOS single-chip 8-bit microcontroller with on-chip EEPROM

80C851/83C851

## OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator, as shown in the logic symbol, page 1085.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

## RESET

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. To insure a good power-up reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few milliseconds) plus two machine cycles. At power-up, the voltage on V<sub>DD</sub> and RST must come up at the same time for a proper start-up.

**Note:** Before entering the idle or power-down modes, the user has to ensure that there is no EEPROM erase/write cycle in progress

(i.e., the EWP bit has to be reset before activating the idle or power-down modes; otherwise EEPROM accesses will be aborted).

## IDLE MODE

In idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

## POWER-DOWN MODE

In the power-down mode, the oscillator is stopped and the instruction to invoke power-down is the last instruction executed. Only the contents of the on-chip RAM and EEPROM are preserved. A hardware reset is the only way to terminate the power-down mode. The control bits for the reduced power modes are in the special function register PCON. Table 3 shows the state of the I/O ports during low current operating modes.

## INTERRUPT SYSTEM

External events and the real-time-driven on-chip peripherals require service by the CPU asynchronous to the execution of any particular section of code. To tie the asynchronous activities of these functions to normal program execution, a multiple-source, two-priority-level, nested interrupt system is provided. Interrupt response latency is from 3µs to 7µs when using a 12MHz crystal. The S83C851 acknowledges interrupt requests from 7 sources as follows:

- INT0 and INT1: externally via pins 12 and 13, respectively,
- Timer 0 and timer 1: from the two internal counters,
- Serial port: from the internal serial I/O port or EEPROM (1 vector).

Each interrupt vectors to a separate location in program memory for its service program. Each source can be individually enabled (the EEPROM interrupt can only be enabled when the serial port interrupt is enabled) or disabled and can be programmed to a high or low priority level. All enabled sources can also be globally disabled or enabled. Both external interrupts can be programmed to be level-activated and are active low to allow "wire-ORing" of several interrupt sources to one input pin.

**Note:** The serial port and EEPROM interrupt flags must be cleared by software; all other flags are cleared by hardware.

**Table 3. External Pin Status During Idle and Power-Down Modes**

MODE	PROGRAM MEMORY	ALE	PSEN	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

## ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>

PARAMETER	RATING	UNIT
Storage temperature range	-65 to +150	°C
Voltage on any other pin to V <sub>SS</sub>	-0.5 to +6.5	V
Input or output DC current on any single I/O pin	±5	mA
Power dissipation (based on package heat transfer limitations, not device power consumption)	1.0	W

## NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
2. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
3. Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V<sub>SS</sub> unless otherwise noted.

# CMOS single-chip 8-bit microcontroller with on-chip EEPROM

80C851/83C851

## DC ELECTRICAL CHARACTERISTICS

 $T_{amb} = 0^{\circ}\text{C to } +70^{\circ}\text{C}$  ( $V_{DD} = 5\text{V} \pm 10\%$ ),  $-40^{\circ}\text{C to } +85^{\circ}\text{C}$  ( $V_{DD} = 5\text{V} \pm 10\%$ ), or  $-40^{\circ}\text{C to } +125^{\circ}\text{C}$  ( $V_{DD} = 5\text{V} \pm 10\%$ ),  $V_{SS} = 0\text{V}$ 

SYMBOL	PARAMETER	PART TYPE	TEST CONDITIONS	LIMITS		UNIT
				MIN	MAX	
$V_{IL}$	Input low voltage, except EA	0 to $+70^{\circ}\text{C}$ $-40$ to $+85^{\circ}\text{C}$ $-40$ to $+125^{\circ}\text{C}$		-0.5 -0.5 -0.5	$0.2V_{DD}-0.1$ $0.2V_{DD}-0.15$ $0.2V_{DD}-0.25$	V
$V_{IL1}$	Input low voltage to EA	0 to $+70^{\circ}\text{C}$ $-40$ to $+85^{\circ}\text{C}$ $-40$ to $+125^{\circ}\text{C}$		-0.5 -0.5 -0.5	$0.2V_{DD}-0.3$ $0.2V_{DD}-0.35$ $0.2V_{DD}-0.45$	V
$V_{IH}$	Input high voltage, except XTAL1, RST	0 to $+70^{\circ}\text{C}$ $-40$ to $+85^{\circ}\text{C}$ $-40$ to $+125^{\circ}\text{C}$		$0.2V_{DD}+0.9$ $0.2V_{DD}+1.0$ $0.2V_{DD}+1.0$	$V_{DD}+0.5$ $V_{DD}+0.5$ $V_{DD}+0.5$	V
$V_{IH1}$	Input high voltage, XTAL1, RST	0 to $+70^{\circ}\text{C}$ $-40$ to $+85^{\circ}\text{C}$ $-40$ to $+125^{\circ}\text{C}$		$0.7V_{DD}$ $0.7V_{DD}+0.1$ $0.7V_{DD}+0.1$	$V_{DD}+0.5$ $V_{DD}+0.5$ $V_{DD}+0.5$	V
$V_{OL}$	Output low voltage, ports 1, 2, 3 <sup>6</sup>		$I_{OL} = 1.6\text{mA}^4$		0.45	V
$V_{OL1}$	Output low voltage, port 0, ALE, PSEN <sup>6</sup>		$I_{OL} = 3.2\text{mA}^4$		0.45	V
$V_{OH}$	Output high voltage, ports 1, 2, 3, ALE, PSEN		$I_{OH} = -60\mu\text{A}$ , $I_{OH} = -25\mu\text{A}$ , $I_{OH} = -10\mu\text{A}$	2.4 $0.75V_{DD}$ $0.9V_{DD}$	V V V	V
$V_{OH1}$	Output high voltage, port 0 in external bus mode <sup>5</sup>		$I_{OH} = -800\mu\text{A}$ , $I_{OH} = -300\mu\text{A}$ , $I_{OH} = -80\mu\text{A}$	2.4 $0.75V_{DD}$ $0.9V_{DD}$	V V V	V
$I_{IL}$	Logical 0 input current, ports 1, 2, 3	0 to $+70^{\circ}\text{C}$ $-40$ to $+85^{\circ}\text{C}$ $-40$ to $+125^{\circ}\text{C}$	$V_{IN} = 0.45\text{V}$		-50 -75 -75	$\mu\text{A}$ $\mu\text{A}$ $\mu\text{A}$
$I_{TL}$	Logical 1-to-0 transition current, ports 1, 2, 3	0 to $+70^{\circ}\text{C}$ $-40$ to $+85^{\circ}\text{C}$ $-40$ to $+125^{\circ}\text{C}$	$V_{IN} = 2.0\text{V}$		-650 -750 -750	$\mu\text{A}$ $\mu\text{A}$ $\mu\text{A}$
$I_{L1}$	Input leakage current, port 0, EA		$0.45\text{V} < V_i < V_{DD}$		$\pm 10$	$\mu\text{A}$
$I_{DD}$	Power supply current: Active mode @ 16MHz <sup>1</sup> Idle mode @ 16MHz <sup>2</sup> Power down mode <sup>3</sup>		See note 7		19 3.7 50	mA mA $\mu\text{A}$
$R_{RST}$	Internal reset pull-down resistor			50	150	k $\Omega$
$C_{IO}$	Pin capacitance		$f = 1\text{MHz}$		10	pF

### NOTES:

- The operating supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 5\text{ns}$ ;  $V_{IL} = V_{SS} + 0.5\text{V}$ ;  $V_{IH} = V_{DD} - 0.5\text{V}$ ; XTAL2 not connected; EA = RST = Port 0 =  $V_{DD}$ .
- The idle mode supply current is measured with all output pins disconnected; XTAL1 driven with  $t_r = t_f = 5\text{ns}$ ;  $V_{IL} = V_{SS} + 0.5\text{V}$ ;  $V_{IH} = V_{DD} - 0.5\text{V}$ ; XTAL2 not connected; EA = Port 0 =  $V_{DD}$ ; RST =  $V_{SS}$ .
- The power-down current is measured with all output pins disconnected; XTAL2 not connected; EA = Port 0 =  $V_{DD}$ ; RST = XTAL1 =  $V_{SS}$ .
- Capacitive loading on Port 0 and Port 2 may cause spurious noise pulses to be superimposed on the LOW level output voltage of ALE, Port 1 and Port 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make a 1-to-0 transition during bus operations. In the worst cases (capacitive loading  $> 100\text{pF}$ ), the noise pulse on the ALE line may exceed 0.8V. In such cases it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.
- Capacitive loading on Port 0 and Port 2 may cause the HIGH level output voltage on ALE and PSEN to momentarily fall below the  $0.9V_{DD}$  specification when the address bits are stabilizing.
- Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:  
Maximum  $I_{OL}$  per Port pin: 10mA  
Maximum  $I_{OL}$  per 8-bit port –  
Port 0: 26mA  
Ports 1, 2, and 3: 15mA  
Maximum total  $I_{OL}$  for all output pins: 71mA  
If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
- See Figures 11 through 14 for  $I_{DD}$  test conditions.



# CMOS single-chip 8-bit microcontroller with on-chip EEPROM

80C851/83C851

## AC ELECTRICAL CHARACTERISTICS<sup>1, 2</sup>

SYMBOL	FIGURE	PARAMETER	16MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
$1/f_{CLCL}$	4	Oscillator frequency			1.2	16	MHz
$t_{LHLL}$	4	ALE pulse width	85		$2t_{CLCL}-40$		ns
$t_{AVLL}$	4	Address valid to ALE low	8		$t_{CLCL}-55$		ns
$t_{LLAX}$	4	Address hold after ALE low	28		$t_{CLCL}-35$		ns
$t_{LLIV}$	4	ALE low to valid instruction in		150		$4t_{CLCL}-100$	ns
$t_{LLPL}$	4	ALE low to PSEN low	23		$t_{CLCL}-40$		ns
$t_{PLPH}$	4	PSEN pulse width	143		$3t_{CLCL}-45$		ns
$t_{PLIV}$	4	PSEN low to valid instruction in		83		$3t_{CLCL}-105$	ns
$t_{PXIX}$	4	Input instruction hold after PSEN	0		0		ns
$t_{PXIZ}$	4	Input instruction float after PSEN		38		$t_{CLCL}-25$	ns
$t_{AVIV}$	4	Address to valid instruction in		208		$5t_{CLCL}-105$	ns
$t_{PLAZ}$	4	PSEN low to address float		10		10	ns
<b>Data Memory</b>							
$t_{RLRH}$	5, 6	RD pulse width	275		$6t_{CLCL}-100$		ns
$t_{WLWH}$	5, 6	WR pulse width	275		$6t_{CLCL}-100$		ns
$t_{RLDV}$	5, 6	RD low to valid data in		148		$5t_{CLCL}-165$	ns
$t_{RHDX}$	5, 6	Data hold after RD	0		0		ns
$t_{RHDZ}$	5, 6	Data float after RD		55		$2t_{CLCL}-70$	ns
$t_{LLDV}$	5, 6	ALE low to valid data in		350		$8t_{CLCL}-150$	ns
$t_{AVDV}$	5, 6	Address to valid data in		398		$9t_{CLCL}-165$	ns
$t_{LLWL}$	5, 6	ALE low to RD or WR low	138	238	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
$t_{AW}$	5, 6	Address to RD or WR	120		$4t_{CLCL}-130$		ns
$t_{QW}$	5, 6	Data setup time before WR	288		$7t_{CLCL}-150$		ns
$t_{QVWX}$	5, 6	Data valid to WR transition	3		$t_{CLCL}-60$		ns
$t_{WHQX}$	5, 6	Data hold after WR	13		$t_{CLCL}-50$		ns
$t_{RLAZ}$	5, 6	RD low to address float		0		0	ns
$t_{WHLH}$	5, 6	RD or WR high to ALE high	23	103	$t_{CLCL}-40$	$t_{CLCL}+40$	ns
<b>External Clock</b>							
$t_{CHCX}$	8	High time	20		20		ns
$t_{CLOX}$	8	Low time	20		20		ns
$t_{CLCH}$	8	Rise time		20		20	ns
$t_{CHCL}$	8	Fall time		20		20	ns
<b>Erase/write timer constant<sup>3</sup></b>							
$t_{E/W}$		Erase/write cycle time	4	20	4	20	ms
$t_E$		Erase time	2	10	2	10	ms
$t_W$		Write time	2	10	2	10	ms
$t_S$		Data retention time <sup>4</sup>	10		10		years
$NE/W$		Erase/write cycles <sup>5</sup>	50,000		50,000		cycles

### NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.
- The power-off fall-time of  $V_{DD}$  must be less than 1ms to prevent an overwrite pulse from being generated in the EEPROM which can cause spurious parasitic writing to EEPROM cells. If the  $V_{DD}$  power-off full-time is greater than 1ms, a power-off reset signal should be generated to prevent this condition from occurring.
- Test condition:  $T_{amb} = +55^{\circ}\text{C}$ .
- Number of erase/write cycles for each EEPROM byte.

## CMOS single-chip 8-bit microcontroller with on-chip EEPROM

80C851/83C851

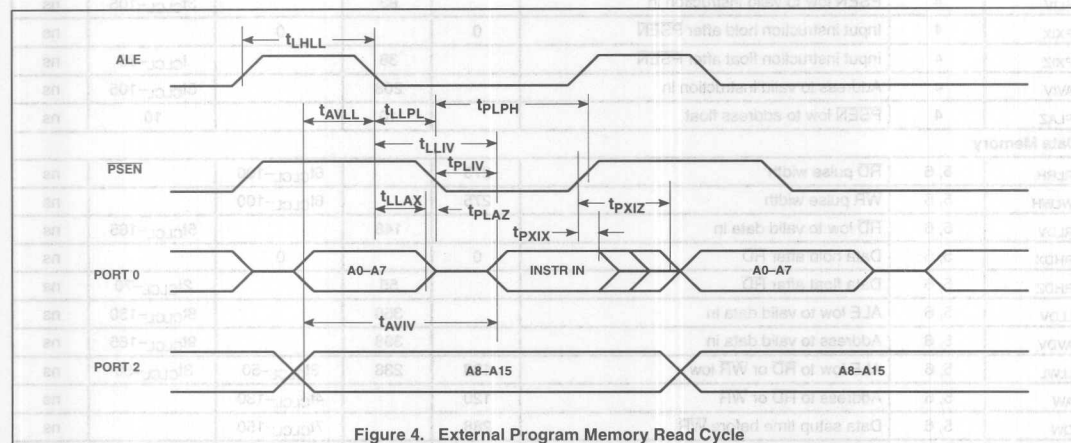
## EXPLANATION OF THE AC SYMBOLS

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

A – Address  
C – Clock  
D – Input data  
H – Logic level high  
I – Instruction (program memory contents)

- L – Logic level low, or ALE
- P – PSEN
- Q – Output data
- R – RD signal
- t – Time
- V – Valid
- W – WR signal
- X – No longer a valid logic level
- Z – Float

**Examples:**  $t_{AVLL}$  = Time for address valid to ALE low.  
 $t_{LLPL}$  = Time for ALE low to PSEN low.



#### Figure 4. External Program Memory Read Cycle

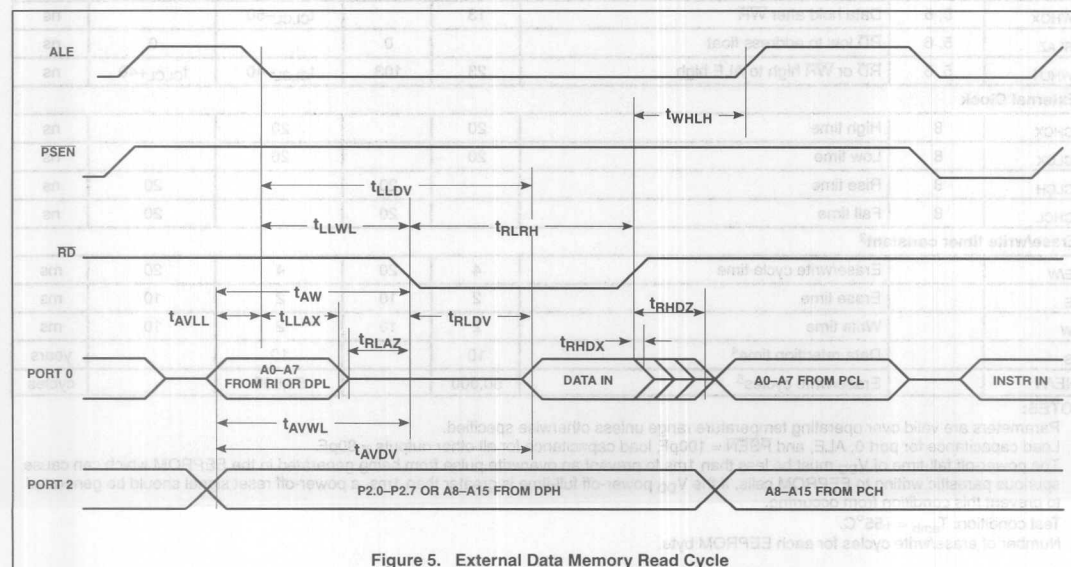
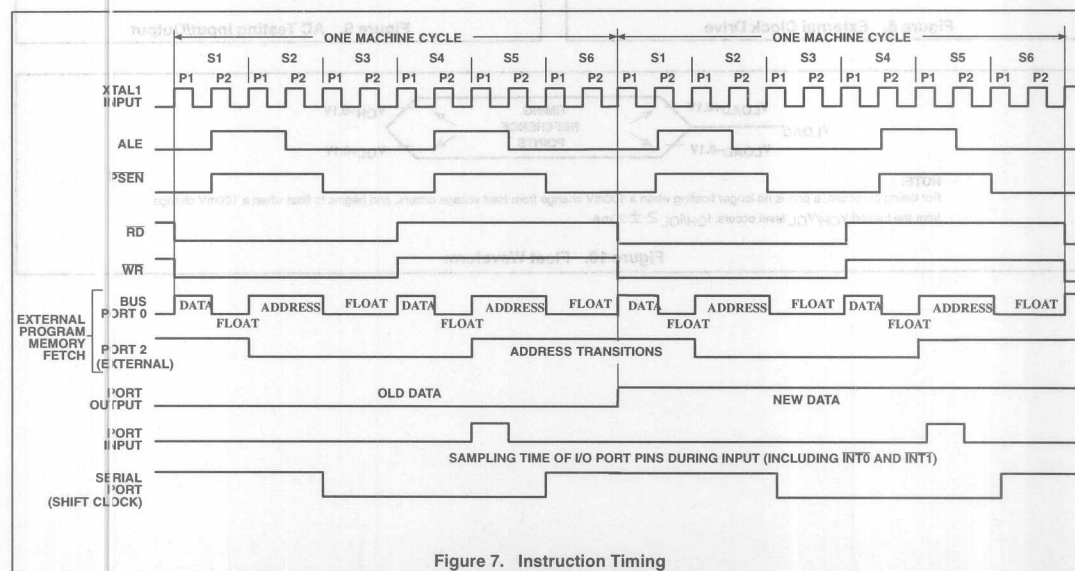
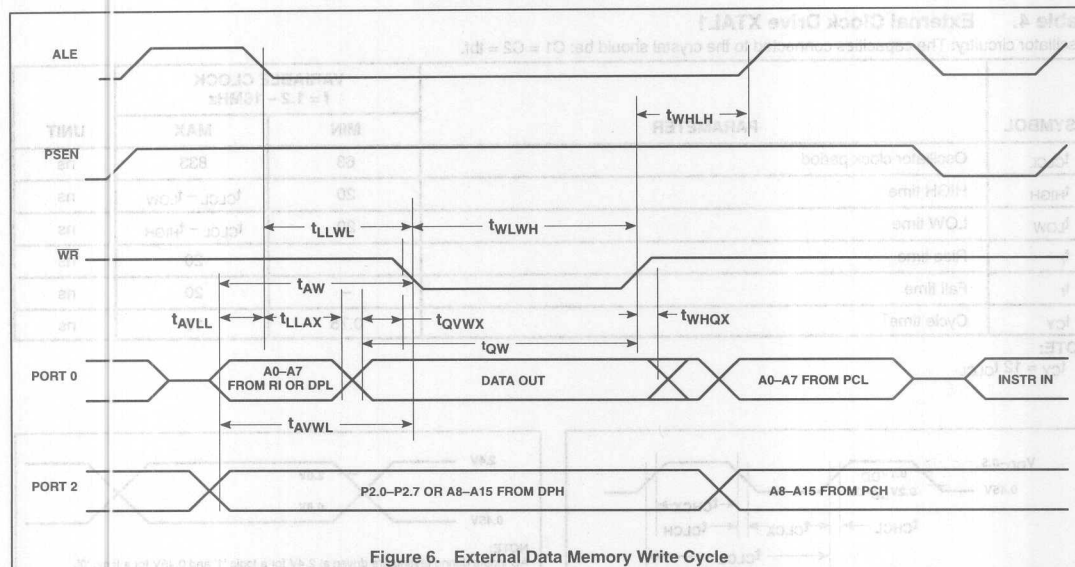


Figure 5. External Data Memory Read Cycle

# CMOS single-chip 8-bit microcontroller with on-chip EEPROM

80C851/83C851



# CMOS single-chip 8-bit microcontroller with on-chip EEPROM

80C851/83C851

**Table 4. External Clock Drive XTAL1**Oscillator circuitry: The capacities connected to the crystal should be:  $C1 = C2 = t_{bf}$ .

SYMBOL	PARAMETER	VARIABLE CLOCK $f = 1.2 - 16\text{MHz}$		UNIT
		MIN	MAX	
$t_{CLCL}$	Oscillator clock period	63	833	ns
$t_{HIGH}$	HIGH time	20	$t_{CLCL} - t_{LOW}$	ns
$t_{LOW}$	LOW time	20	$t_{CLCL} - t_{HIGH}$	ns
$t_r$	Rise time	—	20	ns
$t_f$	Fall time	—	20	ns
$t_{CY}$	Cycle time <sup>1</sup>	0.75	10	ns

NOTE:

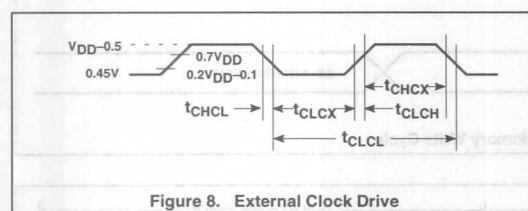
1.  $t_{CY} = 12 t_{CLCL}$ .

Figure 8. External Clock Drive

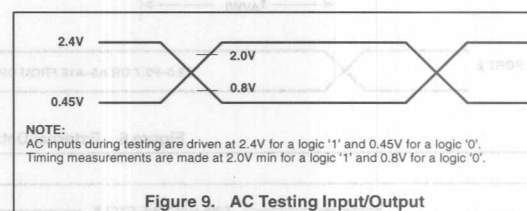


Figure 9. AC Testing Input/Output

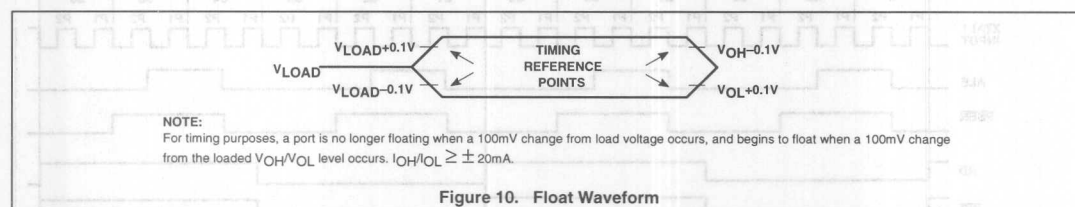
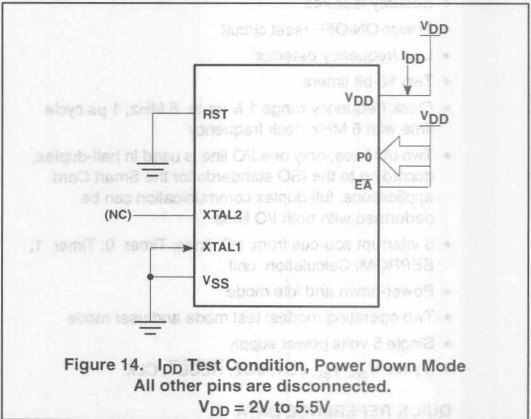
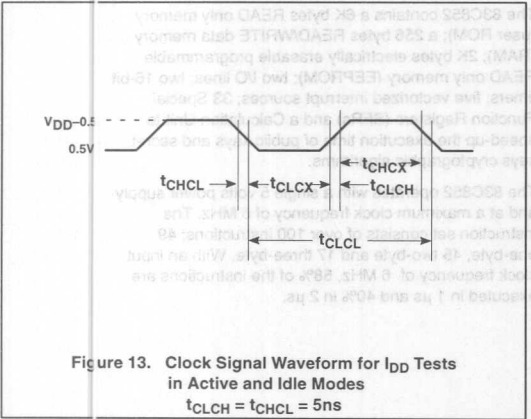
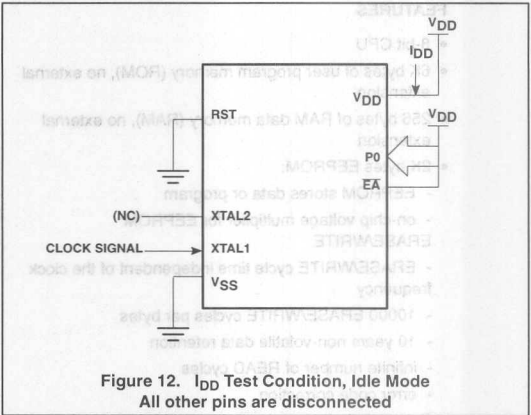
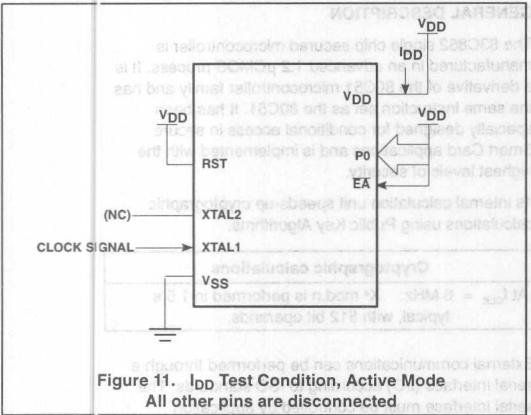


Figure 10. Float Waveform

CMOS single-chip 8-bit microcontroller  
with on-chip EEPROM

80C851/83C851



UNIT	MAX	MIN	CONDITION	PARAMETER	SYMBOL
V	5.5	4.5		supply voltage range	$V_{DD}$
A	10	—	$f_{CLK} = 3.27\text{ MHz}$	supply current	$I_{DD}$
A	20	—	$f_{CLK} = 8.0\text{ MHz}$	supply current operating modes	$I_{DD}$
A	3	—	$f_{CLK} = 0.1\text{ MHz}$	supply current idle mode	$I_{DD}$
W	1	—		total power dissipation	$P_{tot}$
°C	180	-55		storage temperature range	$T_{stg}$
°C	70	0		operating ambient temperature range	$T_{amb}$



## Secured 8-bit microcontroller

83C852

## FEATURES

- 8-bit CPU
- 6K bytes of user program memory (ROM), no external extension
- 256 bytes of RAM data memory (RAM), no external extension
- 2K bytes EEPROM:
  - EEPROM stores data or program
  - on-chip voltage multiplier for EEPROM ERASE/WRITE
  - ERASE/WRITE cycle time independent of the clock frequency
  - 10000 ERASE/WRITE cycles per bytes
  - 10 years non-volatile data retention
  - infinite number of READ cycles
  - error code correction
- Calculation unit for cryptographic calculations
- Security features
- Power-ON/OFF reset circuit
- Low frequency detector
- Two 16-bit timers
- Clock frequency range 1 MHz to 6 MHz; 1  $\mu$ s cycle time with 6 MHz clock frequency
- Two I/O lines; only one I/O line is used in half-duplex, according to the ISO standards for the Smart Card applications; full-duplex communication can be performed with both I/O lines
- 5 interrupt sources from: I/O lines; Timer 0; Timer 1; EEPROM; Calculation unit
- Power-down and idle mode
- Two operating modes: test mode and user mode
- Single 5 volts power supply
- 6 pins:  $V_{DD}$ ,  $V_{SS}$ , I/O1, I/O2, RESET, CLK

## QUICK REFERENCE DATA

SYMBOL	PARAMETER	CONDITION	MIN.	MAX.	UNIT
$V_{DD}$	supply voltage range		4.5	5.5	V
$I_{DD}$	supply current	$f_{CLK} = 3.57$ MHz	—	10	mA
$I_{DD}$	supply current: operating modes	$f_{CLK} = 6.0$ MHz	—	15	mA
$I_{ID}$	supply current: idle mode	$f_{CLK} = 6.0$ MHz	—	3	mA
$P_{tot}$	total power dissipation		—	1	W
$T_{stg}$	storage temperature range		-65	150	°C
$T_{amb}$	operating ambient temperature range		0	70	°C

## GENERAL DESCRIPTION

The 83C852 single chip secured microcontroller is manufactured in an advanced 1.2  $\mu$ COS process. It is a derivative of the 80C51 microcontroller family and has the same instruction set as the 80C51. It has been specially designed for conditional access in secure Smart Card applications and is implemented with the highest levels of security.

Its internal calculation unit speeds-up cryptographic calculations using Public Key Algorithms.

## Cryptographic calculations

At  $f_{CLK} = 6$  MHz:  $X^e \text{ mod } n$  is performed in 1.5 s typical, with 512 bit operands.

External communications can be performed through a serial interface (I/O) according to ISO standards. The serial interface must be controlled by application software for access to the 83C852 internal memory.

The 83C852 contains a 6K bytes READ only memory (user ROM); a 256 bytes READ/WRITE data memory (RAM); 2K bytes electrically erasable programmable READ only memory (EEPROM); two I/O lines; two 16-bit timers; five vectorized interrupt sources; 33 Special Function Registers (SFRs) and a Calculation Unit to speed-up the execution time of public keys and secret keys cryptographic algorithms.

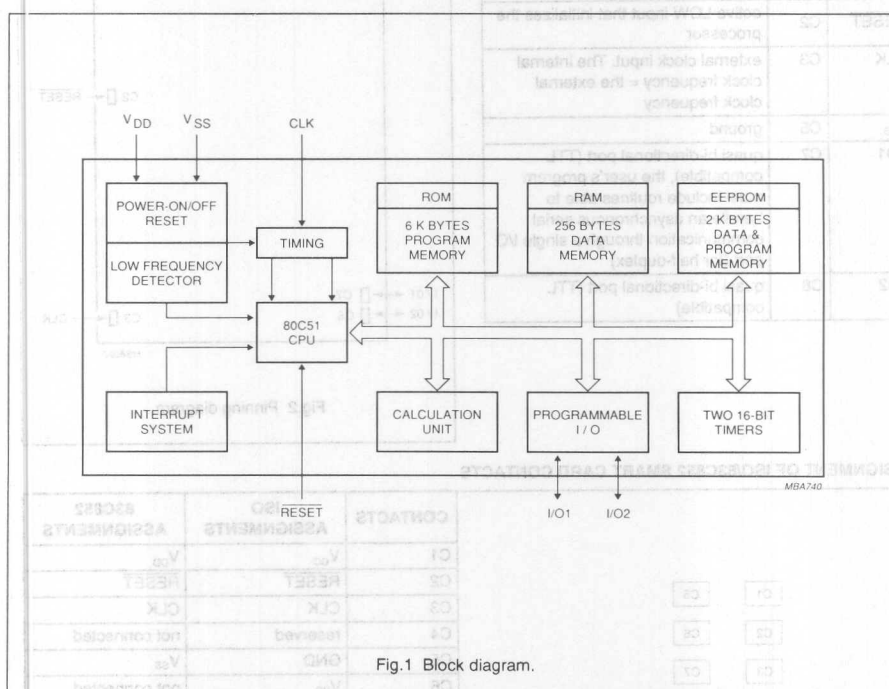
The 83C852 operates with a single 5 volts power supply and at a maximum clock frequency of 6 MHz. The instruction set consists of over 100 instructions: 49 one-byte, 45 two-byte and 17 three-byte. With an input clock frequency of 6 MHz, 58% of the instructions are executed in 1  $\mu$ s and 40% in 2  $\mu$ s.

## Secured 8-bit microcontroller

83C852

## ORDERING INFORMATION

EXTENDED TYPE NUMBER	PACKAGE			
	PADS	PAD POSITION	MATERIAL	CODE
83C852 die	6	X Y coordinates	die	—
83C852P	tbf	tbf	plastic DIL	tbf



Secured 8-bit microcontroller

83C852

PINNING

SYMBOL	PIN (PAD)	DESCRIPTION
V <sub>DD</sub>	C1	+5 volts power supply pin during normal operation, idle mode and power-down mode
RESET	C2	active LOW input that initializes the processor
CLK	C3	external clock input. The internal clock frequency = the external clock frequency
V <sub>SS</sub>	C5	ground
I/O1	C7	quasi bi-directional port (TTL compatible), the user's program must include routines able to handle an asynchronous serial communication through a single I/O port. (for half-duplex)
I/O2	C8	quasi bi-directional port (TTL compatible)

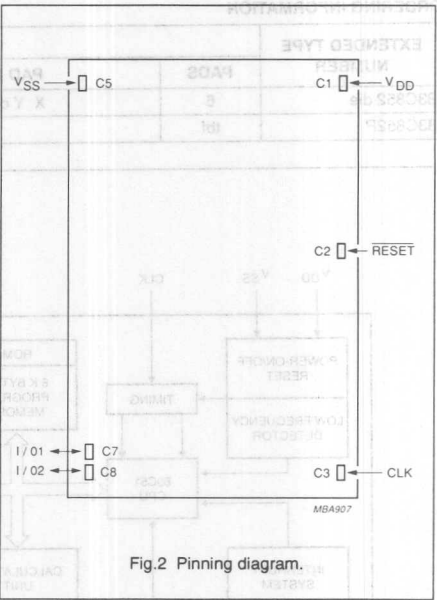


Fig.2 Pinning diagram.

ASSIGNMENT OF ISO/83C852 SMART CARD CONTACTS

CONTACTS	ISO ASSIGNMENTS	83C852 ASSIGNMENTS
C1	V <sub>CC</sub>	V <sub>DD</sub>
C2	RESET	RESET
C3	CLK	CLK
C4	reserved	not connected
C5	GND	V <sub>SS</sub>
C6	V <sub>PP</sub>	not connected
C7	I/O	used for I/O1
C8	reserved	reserved for I/O2

contact assignments are specified in part 2 of ISO 7816

Fig.3 Contact assignments.

## Secured 8-bit microcontroller

## 83C852

## FUNCTIONAL DESCRIPTION

## General

The 83C852 is specially designed for secured applications such as conditional access and transactions in a Smart Card environment. It has a Calculation Unit which makes the microcontroller dedicated to asymmetric crypto systems. Special Function Registers (SFRs) are available to the user to manipulate memory transactions and calculation unit operations.

The address bus of the EEPROM is mixed to prevent fraudulent access and optical scanning. The EEPROM has an hardware error code correction which guarantees the data content integrity. The 83C852 is able to read and modify a part of the internal program memory contained in EEPROM.

The 83C852 has two software selectable modes of reduced activity for further power reduction, the idle mode and power-down mode:

- the idle mode freezes the CPU while allowing the RAM, the timers and the interrupt system to continue functioning.
- the power-down mode saves the RAM content and disables all other chip functions.

The 83C852 has 33 SFRs available for use by the user (see Table 23). The functional descriptions and usage of the SFRs as they co-relate to the RAM, EEPROM and the calculation unit activities, are described within the following sections.

## Memory organisation

The central processing unit (CPU) manipulates operands in three memory spaces, (see Fig. 4) these are:

- 6K-byte internal program memory (ROM)
- 2K-byte program and data memory (EEPROM)
- 256-byte internal data memory (RAM).

The 256-byte internal RAM memory address space is sub-divided into:

- 128-byte internal data RAM locations 00 to 7FH. This address space is accessible with direct and indirect addressing
- 128-byte internal data RAM locations 80H to FFH. This address space is accessible with indirect addressing only

- 128-byte Special Function Register (SFR) address space 80H to FFH. This address space is parallel to the upper 128 byte RAM. It is accessible with direct addressing only. 33 SFRs reside inside this area, the remaining address space in between is unused.

## EEPROM

The EEPROM has a capacity of 2K bytes (words) see Fig. 5. With its built-in error correction hardware the EEPROM is a very reliable non-volatile memory. In addition to each single stored data byte, 4 extra "parity" bits are stored in EEPROM. Single-bit errors per byte are automatically corrected when reading the memory. It can be accessed by both CPU and calculation unit (however, not at the same time).

Programming of the EEPROM is completely controlled by the EEPROM's sequencer. The EEPROM can be used either both as data memory and program memory for the CPU, or as data memory for the calculation unit.

## EEPROM AS DATA MEMORY:

When the CPU executes opcodes from internal ROM (program address < 8000H), the EEPROM can be used as a data memory. The communication between CPU and EEPROM is performed via 6 SFRs (see Table 1), these comprise:

- 2 EEPROM (SFRs) address registers (HIGH and LOW address byte)
- 1 EEPROM (SFR) data register for READ and WRITE operations
- 2 EEPROM (SFRs) control registers to select the various operating and test modes
- 1 EEPROM (SFR) timer register to adapt the ERASE/WRITE time to the operating clock frequency.

## EEPROM AS PROGRAM MEMORY:

When the program counter is higher than 7FFFH, the EEPROM is used as a program memory. The CPU fetches opcodes directly from the EEPROM. **EADRH, EADRL1 and EDAT registers cannot be written.** Their contents in this mode are irrelevant.

Reading data from the EEPROM can still be done with the MOV<sub>C</sub> instruction, but EEPROM write operation is not possible. The EEPROM can only be written by software executed from the ROM area (program address < 8000H).

## Secured 8-bit microcontroller

## 83C852

## EEPROM AS DATA MEMORY FOR THE CALCULATION UNIT:

Communication between calculation unit and EEPROM is performed via SFRs (see Table 1), these comprise:

- 2 EEPROM SFRs address registers (one HIGH and one of two alternate LOW address bytes)
- calculation unit SFRs.

The EEPROM data output is directly connected via a special bus to the data registers of the calculation unit. The calculation unit has direct READ access to the EEPROM.

During memory access, the EEPROM is addressed by one of two address pointers EADRL1 or EADRL2. Both

pointers are loaded by the CPU and decremented by the calculation unit's sequencer. EADRL1 or EADRL2 supply the LOW byte (LSB) of the EEPROM address. The HIGH byte (MSB) of the EEPROM address is taken directly from the EADRH register. The access to the EEPROM from the calculation unit, is controlled by the calculation unit SFRs.

When the access to EEPROM by the calculation unit is active, the EEPROM is not accessible by the CPU, neither as data memory nor as program memory. When the calculation unit is operating, but not accessing the EEPROM, the CPU can READ, WRITE and EXECUTE EEPROM.

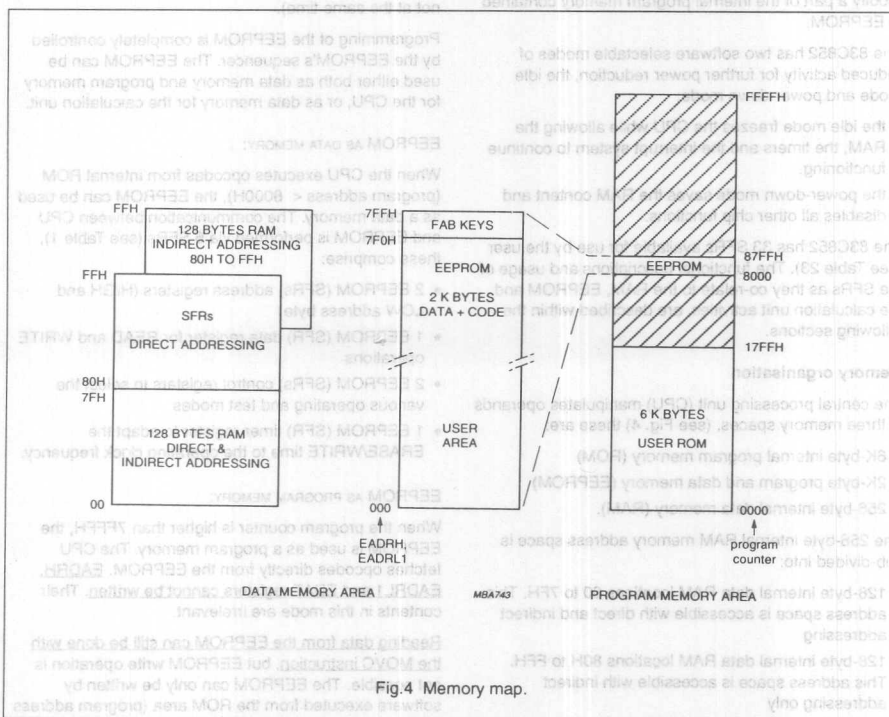


Fig.4 Memory map.



## Secured 8-bit microcontroller

83C852

## EEPROM SFRs

Table 1 provides a listing of the EEPROM associated SFRs:

Table 1 EEPROM SFRs

NAME	SFR ADDRESS	FUNCTION
<b>The communication between the CPU and the EEPROM is performed via the following SFRs:</b>		
EADRL1	0F2H	address register (LSB)
EADRH	0F3H	address register (MSB)
EDAT	0F4H	data register
ETIM	0F5H	EEPROM timer register
ENCTRL1	0F6H	control register for normal operation modes
ENCTRL2	0F7H	control register for special test modes
<b>The communication between the calculation unit and the EEPROM is performed via the following SFRs:</b>		
EADRL2	0F1H	address pointer 2: LSB of the EEPROM address (WRITE: reload EADRL2 register; READ: read counter)
EADRL1	0F2H	address pointer 1: LSB of the EEPROM address (WRITE: reload EADRL1 register; READ: read counter)
EADRH	0F3H	address register (MSB)
Two address pointers (SFRs) are necessary for EEPROM access by the calculation unit. SFRs EADRH and either EADRL1 or EADRL2		

## EEPROM SFR descriptions

## EADRH

EADRH is an 8-bit register (SFR), used as an address pointer, it is loaded from the CPU and contains the highest byte (MSB) of the EEPROM address. Only bits EADRH.7, 2, 1, 0 are relevant. Default value after reset is '10000000B'.

## EADRL1, EADRL2

EADRL1 is an 8-bit register (SFR), used as an address pointer, it is loaded from the CPU. Its contents are transferred into an 8-bit down counter which provides the LOW byte (LSB) of the EEPROM address. The CPU has WRITE access to the EADRL1 register and READ access to the down counter.

Default value after reset of both the EADRL1 register and associated down counter are 00H. The transfer of EADRL1 to the down counter and the decrement are controlled by the calculation unit. The behaviour of EADRL1 depends on whether or not there is an access in progress from the calculation unit to the EEPROM.

No access from the calculation unit to the EEPROM:

- when there is no active access to the EEPROM from the calculation unit, the contents of the EADRL1 register are continuously loaded into its associated down counter. There is no hardware-decrement of the down counter. The combination of EADRL1 plus its down counter behaves therefore as a normal register. In this mode, the LOW address byte of the EEPROM is always supplied by EADRL1. EADRL2 behaves similar to EADRL1 but it is not used for EEPROM addressing. Its contents are irrelevant.

## Calculation unit access to the EEPROM:

- EADRL1 is the EEPROM LOW address pointer used to address an operand (Ai) stored in the EEPROM. At the beginning of the calculation unit's computation cycle, the address content of the EADRL1 is loaded into its associated down counter. During the calculation, further EADRL1 address transfers to the down counter stop, whilst the down counter is decremented by the calculation unit's sequencer. During a calculation, the EADRL1 register can be reloaded from the CPU with a new address. This new address will then be used during the next calculation.

- EADRL2 (SFR) is the second EEPROM LOW address pointer which is required for some operations of the calculation unit. It is used to address an operand (Xi) stored in the EEPROM. The function of EADRL2 is similar to EADRL1 function. EADRL2 cannot be used as a second address register for normal CPU access to the EEPROM.

#### Data register EDAT

This register (SFR) is used to read data from the currently addressed EEPROM byte. When EDAT is written during BYTE MODE, its contents will be programmed into the addressed EEPROM byte. When EDAT is written during ROW ERASE or BLOCK ERASE mode, the ROW ERASE or BLOCK ERASE operation is

started. In this mode, the data written to EDAT is then irrelevant. The ECNTL1 status bits EWP and IFE indicate whether the EEPROM ERASE/WRITE operation is still active. Whilst the EEPROM programming is in progress, rewriting data to EDAT is not allowed.

#### Timer register ETIM

The ETIM timer register (SFR) is required to adapt the ERASE/WRITE time to the clock frequency. ERASE ( $t_e$ ) and WRITE ( $t_w$ ) times of 5 ms each are required. The user has to ensure that the ERASE or WRITE time is neither too short nor too long. Table 2 gives values for ETIM register for given clock frequencies. ETIM has to be loaded by software in advance to the first ERASE/WRITE operation. ETIM's default value after reset is '08H'.

**Table 2** ETIM timer values

The general formula is: Value (decimal) =  $(f_{CLK} \text{ kHz}/102.4) - 2$

OPERATING $f_{CLK}$ MHz	VALUES FOR ETIM			
	BINARY		HEXADECIMAL	DECIMAL
	MSB	LSB		
1.0	0000	1000	8	8
2.0	0001	0010	12	18
3.0	0001	1011	1B	27
3.57	0010	0001	21	33
4.0	0010	0101	25	37
4.92	0010	1110	2E	46
5.0	0010	1111	2F	47
6.0	0011	1001	39	57

Secured 8-bit microcontroller

83C852

Control register ECNTRL1

ECNTRL1 is the control register (SFR) for the several user operation modes of the EEPROM.

Table 3 ECNTRL1 SFR

IFE	EEINT	EWP	-	3	2	1	0
7	6	5	4				
			BYTE MODE →	0	0	0	0
			ROW ERASE →	1	1	0	0
			BLOCK ERASE →	1	0	1	0
			TEST MODE →	1	1	1	1

Table 4 Description of the ECNTRL1 bits

SYMBOL / PARAMETER	FUNCTION
<b>ECNTRL1.7</b>	
FE, interrupt flag EEPROM	set by the EEPROM sequencer after completion of an EEPROM WRITE access, or set and reset by software. When (IFE is set 1 and EEINT is set 1), an interrupt request is done. Interrupt vector 0023H will be forced if the bits EA and EE inside the interrupt Enable Register IE are also set 1.
<b>ECNTRL1.6</b>	
EEINT, enable EEPROM interrupt	set and reset by software. Enables an EEPROM interrupt request when HIGH.
<b>ECNTRL1.5</b>	
EWP, ERASE/WRITE in progress	set and reset by the EEPROM sequencer. EWP is active HIGH during EEPROM write operations. Consecutive write operations to EDAT are not allowed as long as EWP is set. EWP cannot be set or reset by software.
<b>ECNTRL1.4</b>	
-	reserved.
<b>ECNTRL1 bits 3, 2, 1, 0</b>	
operation mode select: BYTE MODE (0000)	normal E <sup>2</sup> PROM mode, default mode after reset. In this mode READ or WRITE access to one byte at a time is possible.
READ mode	This is the default mode when BYTE MODE is selected. The contents of the addressed byte are available in the data register EDAT.
WRITE mode	This mode is activated after loading of the data register EDAT with the data byte to be written. Before writing EDAT, the address registers EADRL1 and EADRH must be loaded first. Depending on the previous contents of the addressed memory cells, the EEPROM sequencer decides whether to do a WRITE cycle ( $t_w$ ), or a combined ERASE/WRITE cycle ( $t_e + t_w$ ). A WRITE cycle is carried out when the previous memory content has been 00H. Otherwise an ERASE/WRITE cycle is carried out.

## Secured 8-bit microcontroller

83C852

SYMBOL / PARAMETER	FUNCTION
operation mode select: ROW ERASE (1100)	in this mode the contents of the addressed memory row will be erased. The three LSB's of EADRL1 are not significant, i.e. 8 bytes addressed by EADRL1 will be cleared in the same time normally needed to clear one single byte ( $t_{\text{ROW ERASE}} = t_e$ ). The ROW ERASE operation can be started by writing the EDAT register. The data that is written to EDAT is not significant. Writing of the erased 8 memory cells then takes only 8 WRITE cycles. This is much faster than writing 8 data bytes without a previous ROW ERASE. Such an operation would have taken a total = $8 t_e + 8 t_w$ .
operation mode select: BLOCK ERASE (1010)	in this mode all memory cells of the EEPROM will be cleared. The BLOCK ERASE operation can be started by writing EDAT. The contents of the data and address registers EDAT, EADRL1, EADRH are don't care.
operation mode select: TEST MODE (1111)	the selection of a specific EEPROM TEST MODE within the ECNTRL2 register is only possible when the ECNTRL1 register is switched in advance to TEST MODE.

## Control register ECNTRL2

ECNTRL2 is the control register (SFR) for the several test modes of the EEPROM.

Table 5 ECNTRL2 SFR

READ ONLY				READ/WRITE			
SP3	SP2	SP1	SP0	TM3	TM2	TM1	TM0
7	6	5	4	3	2	1	0
NO TEST →				0	0	0	0
READ EEPROM WITHOUT ERRORCORRECTION →				1	0	1	1

Table 6 Description of the ECNTRL2 bits

SYMBOL / PARAMETER	FUNCTION
ECNTRL2 bits 7, 6, 5, 4	
SP3, SP2, SP1, SP0	this part of the ECNTRL2 register is READ only. The upper 4 bits of ECNTRL2 carry either the syndrome word which is generated by the EEPROM error correction logic or the parity bits stored in parallel to the data word in EEPROM memory. The syndrome word is always output during BYTE MODE READ (EWP = 0). A value of '0000B' means that no error has been detected/corrected. The parity bits are output during READ EEPROM WITHOUT ERRORCORRECTION TESTMODE or while EWP = 1.
ECNTRL2 bits 3, 2, 1, 0	
TM3, TM2, TM1, TM0	this part of ECNTRL2 register is READ/WRITE. The lower 4 bits of ECNTRL2 are used to select one of the EEPROM test-modes. The selection of any test-mode is only possible if ECNTRL1 has been set to 'XXXX1111B' before. Otherwise TM3, TM2, TM1, TM0 are held at '0000B'.

## Secured 8-bit microcontroller

83C852

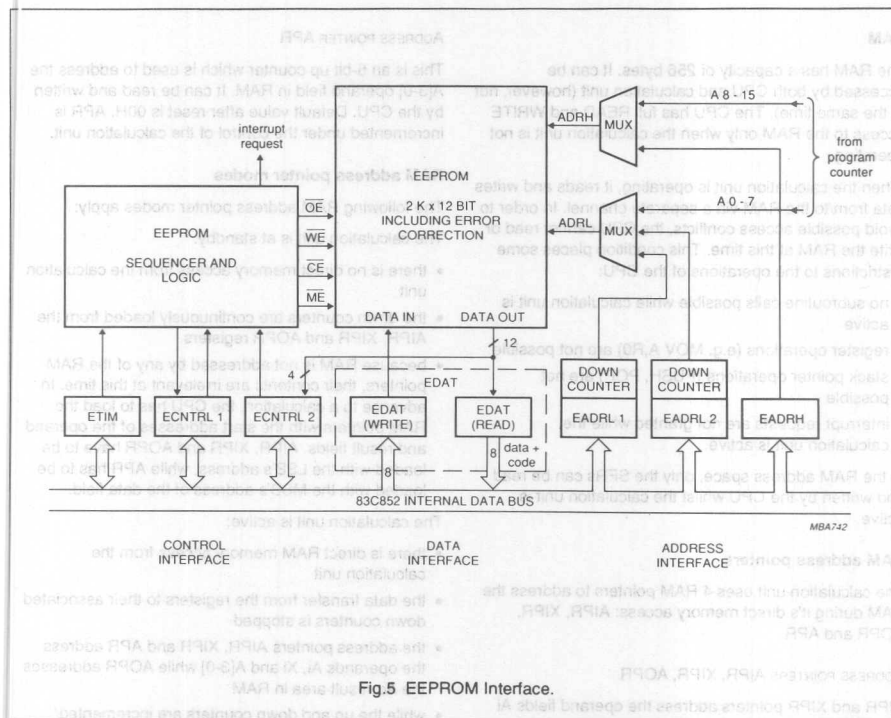


Fig.5 EEPROM Interface



## Secured 8-bit microcontroller

83C852

**RAM**

The RAM has a capacity of 256 bytes. It can be accessed by both CPU and calculation unit (however, not at the same time). The CPU has full READ and WRITE access to the RAM only when the calculation unit is not operating.

When the calculation unit is operating, it reads and writes data from/to the RAM via a separate channel. In order to avoid possible access conflicts, the CPU cannot read or write the RAM at this time. This condition places some restrictions to the operations of the CPU:

- no subroutine calls possible while calculation unit is active
- register operations (e.g. MOV A,R0) are not possible
- stack pointer operations (PUSH, POP) are not possible
- interrupt requests are not granted while the calculation unit is active.

In the RAM address space, only the SFRs can be read and written by the CPU whilst the calculation unit is active.

**RAM address pointers**

The calculation unit uses 4 RAM pointers to address the RAM during its direct memory access: AIPR, XIPR, AOPR and APR.

**ADDRESS POINTERS AIPR, XIPR, AOPR**

AIPR and XIPR pointers address the operand fields Ai resp. Xi inside RAM, while AOPR addresses a RAM area Ao where the calculation result is to be stored. Each pointer consists of an 8-bit register with associated 8-bit down counter. The counter provides a RAM address. It is parallel loaded from its register.

The CPU has WRITE access to the registers and READ access to the counters. Default values after reset for all registers and counters are 00H. The data transfer from the registers to their down counters and the counter decrement are controlled by the calculation unit.

**ADDRESS POINTER APR**

This is an 8-bit up counter which is used to address the A[3-0] operand field in RAM. It can be read and written by the CPU. Default value after reset is 00H. APR is incremented under the control of the calculation unit.

**RAM address pointer modes**

The following RAM address pointer modes apply:

The calculation unit is at standby:

- there is no direct memory access from the calculation unit
- the down counters are continuously loaded from the AIPR, XIPR and AOPR registers
- because RAM is not addressed by any of the RAM pointers, their contents are irrelevant at this time. In advance to a calculation, the CPU has to load the RAM pointers with the start addresses of the operand and result fields. AIPR, XIPR and AOPR have to be loaded with the LSB's address, while APR has to be loaded with the MSB's address of the data field.

The calculation unit is active:

- there is direct RAM memory access from the calculation unit
- the data transfer from the registers to their associated down counters is stopped
- the address pointers AIPR, XIPR and AOPR address the operands Ai, Xi and A[3-0] while AOPR addresses the Ao result area in RAM
- while the up and down counters are incremented/decremented under control of the calculation unit, the CPU can reload the registers with new addresses, ready for transfer to their down counters at the beginning of the next calculation cycle.

Secured 8-bit microcontroller

83C852

CALCULATION UNIT

This unit computes, with it's associated software, any exponential functions like  $x^e \text{ mod } n$ . It has been designed to optimize the calculation time of exponent modulo N. It uses 196 bytes of RAM for 512-bit length operands.

The calculation unit does not carry out a complete exponentiation in one step. However, it provides a set of basic instructions, from which the complete exponentiation algorithm can be built by a dedicated software. All of these basic instructions operate on data-fields inside RAM and EEPROM. The width of these data-fields is variable. A typical operand width is 512 bits.

CALCULATION UNIT PERFORMANCE

At  $f_{CLK} = 6 \text{ MHz}$ :  $X^e \text{ modulo } N$  is performed in 1.5 s typical, with 512 bit operands.

The basic operation of the calculation unit is to multiply either a 24-bit number or a 32-bit number with a long-word (e.g. 512-bit) and adding the result to another long-word. Further XOR and shift operations may be carried out to give the final result. With a 32-bit number this operation completes in typically 45  $\mu\text{s}$  at 6 MHz clock frequency.

To reach this speed, the calculation unit's architecture provides:

- fast multiplication and addition
- fast carry handling
- fast data transfers to fetch operands from RAM or EEPROM and to store results in RAM
- simultaneous operation of both CPU and calculation unit.

CALCULATION UNIT BASIC OPERATION

The basic operation of the calculation unit is:	
$B = [(A + a * X) \oplus \text{value}] * 2^n$	Where "A","B" and "X" are large numbers, "a" is a 3 or 4 byte part of the large number "Y" and "value" is either one same byte used for each result byte or the large number "X". "n" is the number of bit-shifts for the calculation result, n can be either 0 or 32.
Operation set of the calculation unit:	
$B = (A + a * X) \oplus \text{value} * 2^0$	multiply and accumulate step (see Fig. 6).
$B = (0 + a * 0) \oplus \text{value} * 2^0$	memory initialization (see Fig. 7).
$A = (A + a * 0) \oplus \text{value} * 2^{32}$	4 byte shift (see Fig. 8).
$B = (A + a * 0) \oplus \text{value} * 2^0$	memory transfer (see Fig. 9).
$B = (A + a * X) \oplus \text{value} * 2^{32}$	multiply, accumulate step with shift (see Fig. 10).
Force $A = 0$ or $X = 0$ and shift depend on the contents of the registers CMD and CMDSTAT.	
A calculation example of $D = M * C \text{ Mod } N$ is shown in Fig.11, where D, M, C and N are large numbers of n-bit length.	

## Secured 8-bit microcontroller

83C852

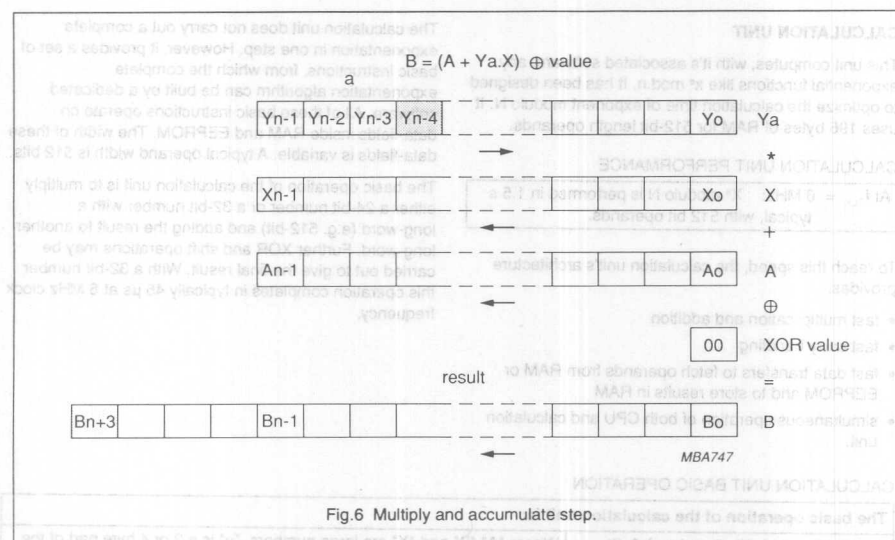


Fig.6 Multiply and accumulate step.

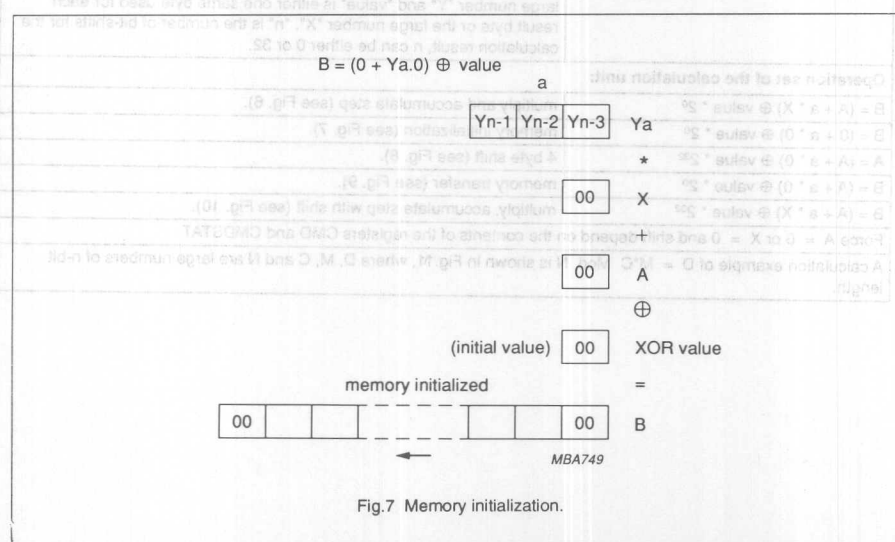
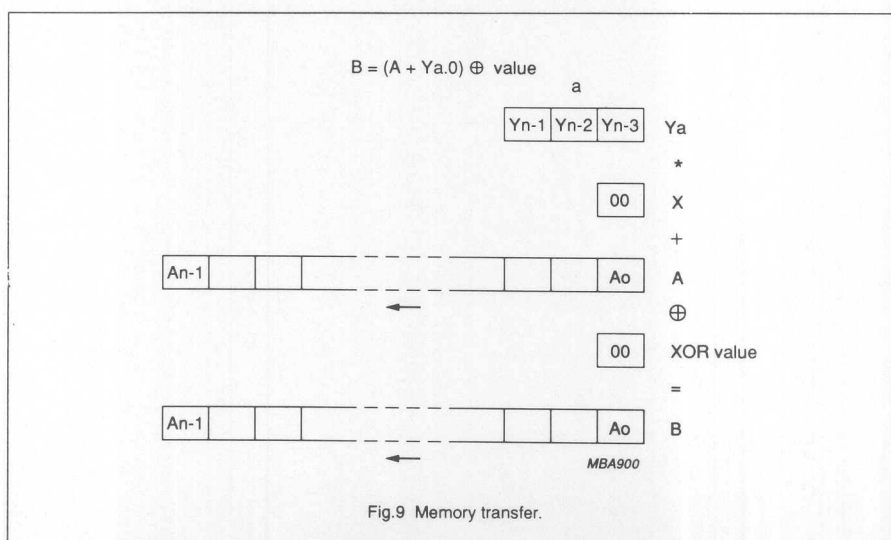
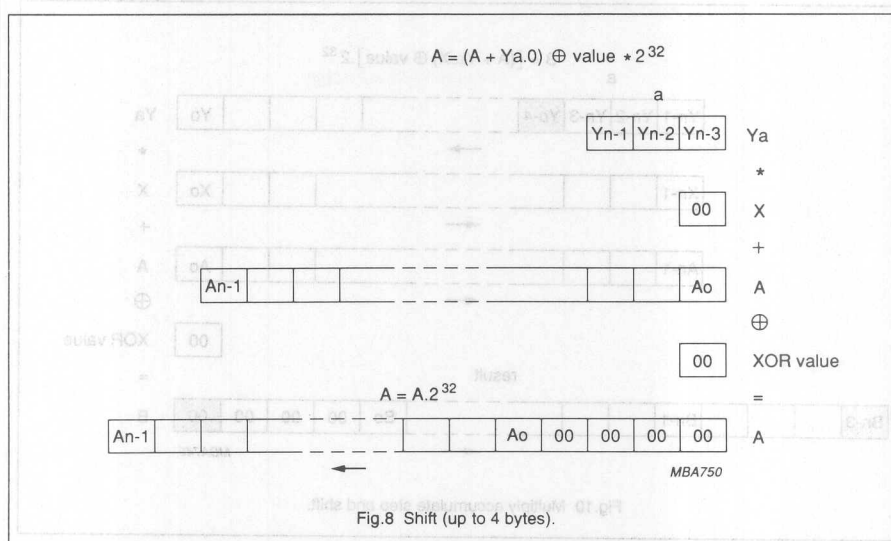


Fig.7 Memory initialization.

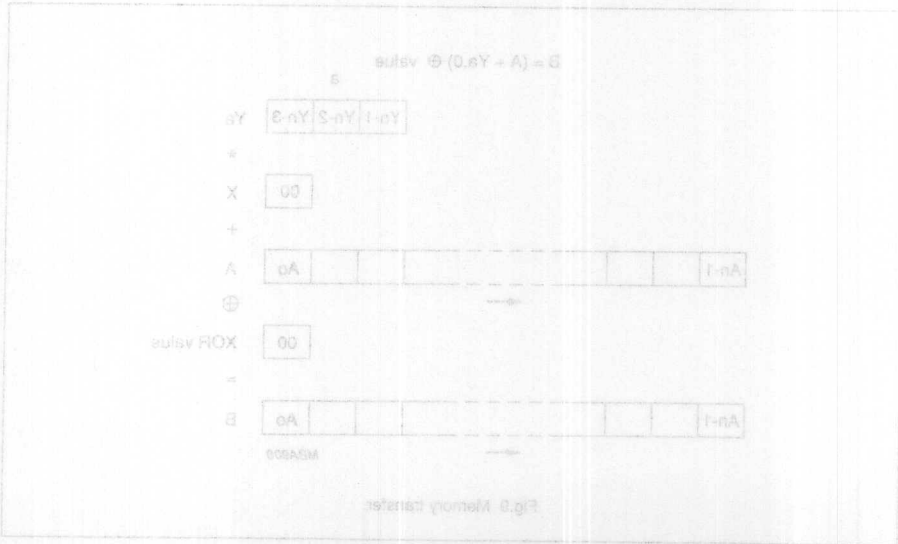
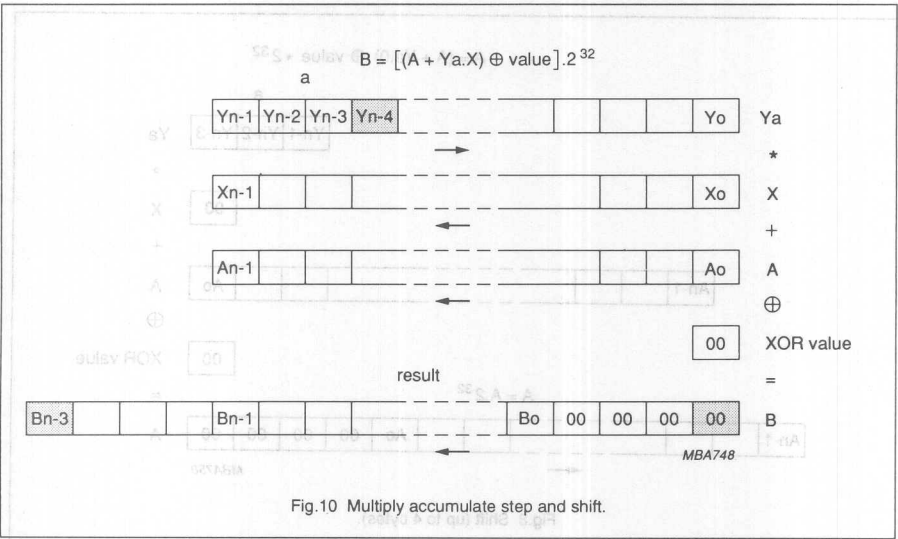
## Secured 8-bit microcontroller

83C852



Secured 8-bit microcontroller

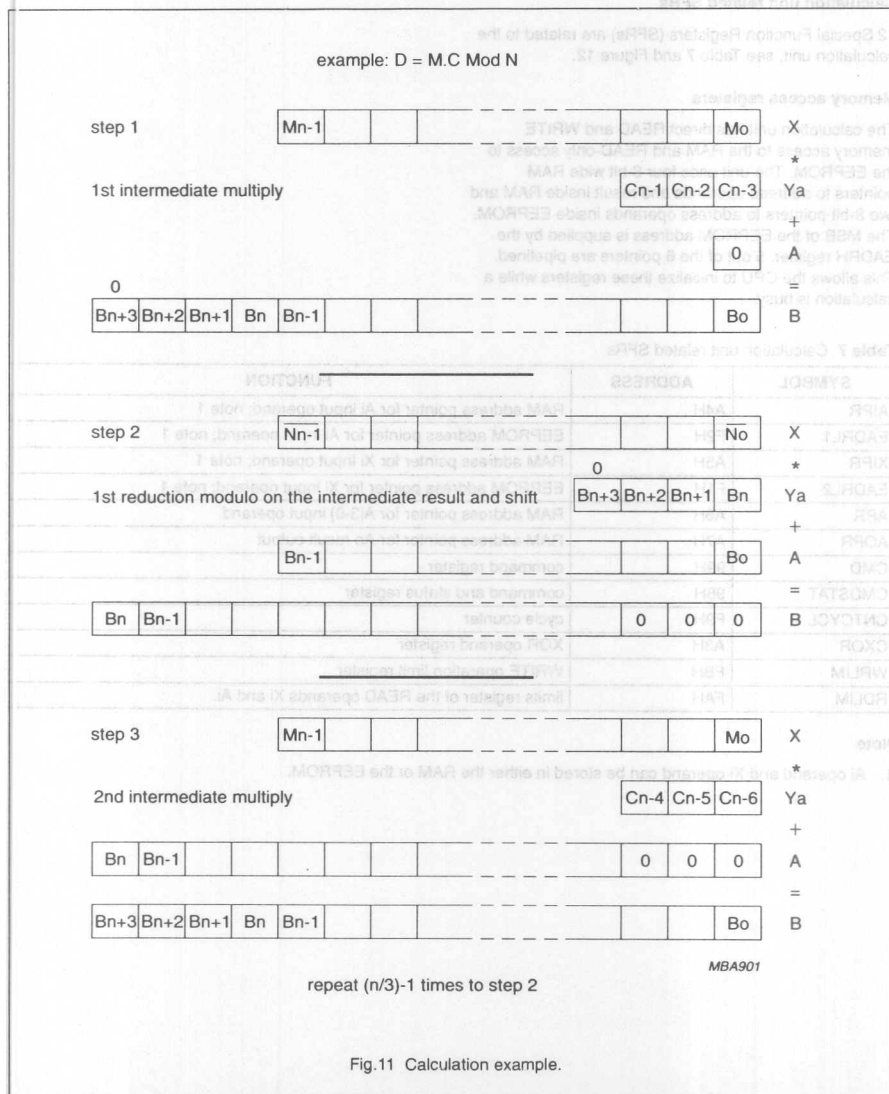
8-bit microcontroller 83C852





## Secured 8-bit microcontroller

83C852



## Secured 8-bit microcontroller

83C852

## Calculation unit related SFRs

12 Special Function Registers (SFRs) are related to the calculation unit, see Table 7 and Figure 12.

## Memory access registers

The calculation unit has direct READ and WRITE memory access to the RAM and READ-only access to the EEPROM. The unit uses four 8-bit wide RAM pointers to address operands and result inside RAM and two 8-bit pointers to address operands inside EEPROM. The MSB of the EEPROM address is supplied by the EADRH register. 5 out of the 6 pointers are pipelined. This allows the CPU to initialize these registers while a calculation is busy.

Table 7 Calculation unit related SFRs

SYMBOL	ADDRESS	FUNCTION
AIPR	A4H	RAM address pointer for Ai input operand; note 1
EADRL1	F2H	EEPROM address pointer for Ai input operand; note 1
XIPR	A5H	RAM address pointer for Xi input operand; note 1
EADRL2	F1H	EEPROM address pointer for Xi input operand; note 1
APR	A6H	RAM address pointer for A[3-0] input operand
AOPR	A7H	RAM address pointer for Ao result output
CMD	99H	command register
CMDSTAT	98H	command and status register
CNTCYCL	F9H	cycle counter
CXOR	A3H	XOR operand register
WRLIM	FBH	WRITE operation limit register
RDLIM	FAH	limits register of the READ operands Xi and Ai.

## Note

1. Ai operand and Xi operand can be stored in either the RAM or the EEPROM.

## Secured 8-bit microcontroller

## 83C852

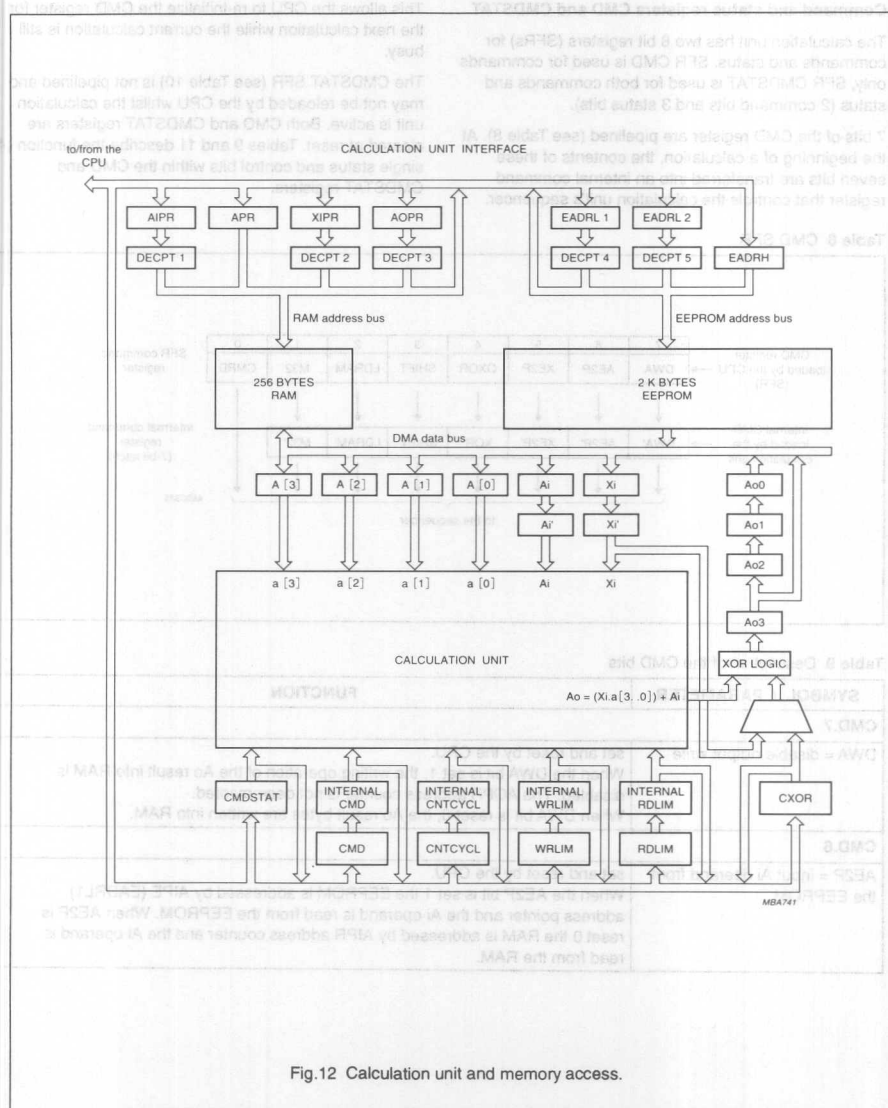


Fig.12 Calculation unit and memory access.

Secured 8-bit microcontroller

80C852

Command and status registers CMD and CMDSTAT

The calculation unit has two 8 bit registers (SFRs) for commands and status. SFR CMD is used for commands only, SFR CMDSTAT is used for both commands and status (2 command bits and 3 status bits).

7 bits of the CMD register are pipelined (see Table 8). At the beginning of a calculation, the contents of these seven bits are transferred into an internal command register that controls the calculation unit's sequencer.

This allows the CPU to re-initialize the CMD register for the next calculation while the current calculation is still busy.

The CMDSTAT SFR (see Table 10) is not pipelined and may not be reloaded by the CPU whilst the calculation unit is active. Both CMD and CMDSTAT registers are cleared at reset. Tables 9 and 11 describe the function of single status and control bits within the CMD and CMDSTAT registers.

Table 8 CMD SFR

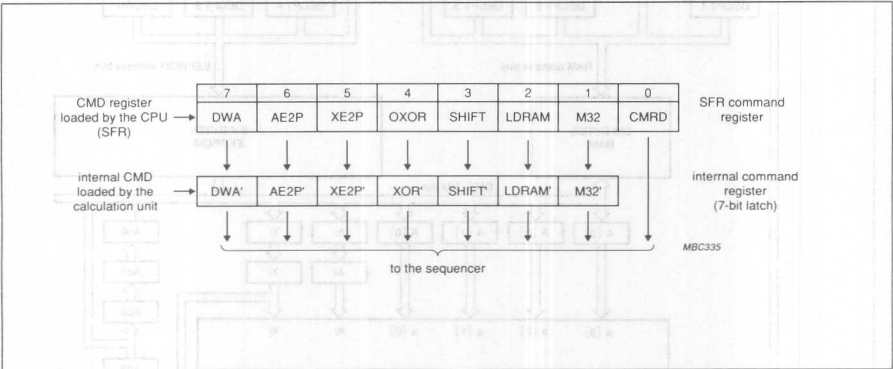


Table 9 Description of the CMD bits

SYMBOL / PARAMETER	FUNCTION
<b>CMD.7</b>	
DWA = disable output write	set and reset by the CPU. When the DWA bit is set 1, the writing operation of the Ao result into RAM is disabled. The AOPR address counter is not decremented. When DWA bit is reset 0, the Ao result bytes are written into RAM.
<b>CMD.6</b>	
AE2P = input Ai operand from the EEPROM	set and reset by the CPU. When the AE2P bit is set 1 the EEPROM is addressed by AIPE (EADRL1) address pointer and the Ai operand is read from the EEPROM. When AE2P is reset 0 the RAM is addressed by AIPR address counter and the Ai operand is read from the RAM.

## Secured 8-bit microcontroller

83C852

SYMBOL / PARAMETER	FUNCTION
<b>CMD.5</b>	
XE2P = input Xi operand from the EEPROM	set and reset by the CPU. When the XE2P bit is set 1 the EEPROM is addressed by XIPE (EADRL2) address pointer and the Xi operand is read from the EEPROM. When XE2P is reset 0 the RAM is addressed by XIPR address counter and the Xi operand is read from the RAM.
<b>CMD.4</b>	
OXOR = output exclusive OR	set and reset by the CPU. When the OXOR bit is set 1 the output data value is: $Xi \oplus$ calculation result. When the OXOR bit is reset 0 the output data value is: CXOR register content $\oplus$ calculation result. If the output value must be the real calculation result: OXOR bit is reset 0 and the CXOR register content = 00H.
<b>CMD.3</b>	
SHIFT = control of Ao result shift	set and reset by the CPU. When the SHIFT bit is set 1 the Ao result consists of four registers Ao3, Ao2, Ao1, Ao0, thus the output data is delayed four times which leads to a 32-bit result shift. If the SHIFT bit is reset 0, the output pipeline has only one register Ao3. The result is not shifted.
<b>CMD.2</b>	
LDRAM = Load A3, A2, A1, A0 registers from the RAM	set and reset by the CPU. When the LDRAM bit is set 1 the calculation unit reloads A3, A2, A1, A0 registers from the RAM at the start of the computation. When the LDRAM bit is reset 0, the A3, A2, A1, A0 registers are reloaded from the Ao3, Ao2, Ao1, Ao0 output pipeline registers.
<b>CMD.1</b>	
M32 = 32-bit operands	set and reset by the CPU. When the M32 bit is set 1 the multiplier operands are A[3..0] (32-bits), APR is incremented by four. When the M32 bit is reset 0 the multiplier operands are A[2..0] (28-bits), APR is incremented by three.
<b>CMD.0</b>	
CMRD = command ready	set by the CPU and cleared by the calculation unit (see Fig. 13). To start a calculation, the CMRD bit is set 1 by the CPU. The CMRD bit will be reset 0 by hardware immediately after beginning the calculation. The CMD register is then ready to take the next command word from the CPU. When the CMRD bit is set 1 by the CPU while the calculation unit is still active, the calculation unit does not stop at the end of the current computation. The new calculation starts immediately with the new command parameters. In this case there will be no interrupt request.



## Secured 8-bit microcontroller

## 83C852

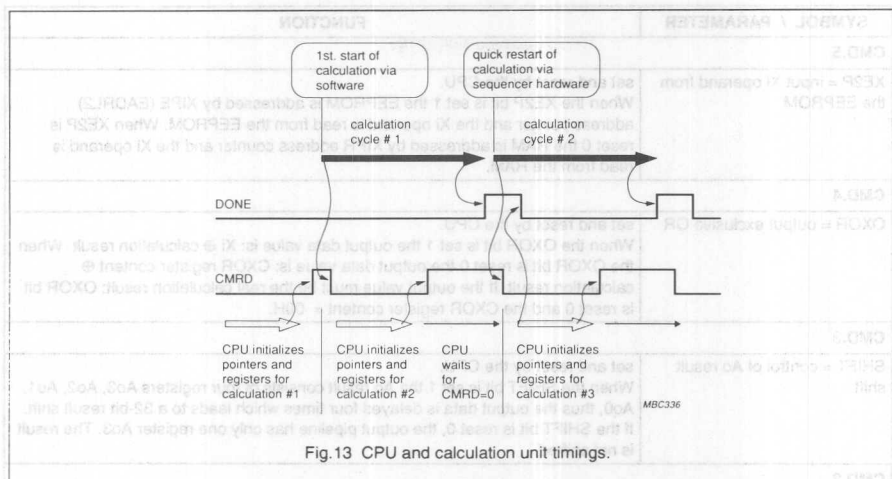


Table 10 CMDSTAT SFR

SFR Command and Status register (SFR) loaded and read by the CPU →				
4	3	2	1	0
DRA	DRX	RUN	CCY	DONE
↓	↓	↑	↑	↑
to the sequencer		from the calculation unit		

Table 11 Description of the CMDSTAT bits

SYMBOL / PARAMETER	FUNCTION
CMDSTAT bits 7, 6, 5, not applicable	
<b>CMDSTAT.4</b>	
DRA = READ disable of the operand Ai	set and reset by the CPU. When the DRA bit is set 1, READ operation of the operand Ai is disabled and Ai is cleared. In this mode the AIPR address counter is not decremented.
<b>CMDSTAT.3</b>	
DRX = READ disable of the operand Xi	set and reset by the CPU. When the DRX bit is set 1, READ operation of the operand Xi is disabled and Xi is cleared. In this mode the XIPR address counter is not decremented.
<b>CMDSTAT.2</b>	
RUN = active calculation unit	set and reset by the calculation unit. While the calculation unit is active, the RUN bit is set 1, otherwise the RUN bit is reset 0 (cleared).

## Secured 8-bit microcontroller

83C852

SYMBOL / PARAMETER	FUNCTION
CMDSTAT bits 7, 6, 5, not applicable	
<b>CMDSTAT.1</b>	
CCY = carry	set and reset by the calculation unit. When the contents of Ao3, Ao2, Ao1, Ao0 output pipeline registers are greater than 0 (zero), the CCY bit is set 1, otherwise CCY is reset 0.
<b>CMDSTAT.0</b>	
DONE = end of the computation	set by the calculation unit, reset by the CPU. The DONE bit is set 1 by the calculation unit at the end of computation. When set 1, an interrupt request is generated. The DONE flag has to be reset 0 by the CPU during an interrupt service routine.

**Cycle counter CNTCYCL**

This is an 8-bit register with associated down counter. It counts the number of bytes of the result which the calculation unit shall carry out. The first byte of the result is always zero.

The CPU has WRITE access to the register and READ access to the down counter. CNTCYCL is cleared to 00H during reset. As long as the calculation unit is in standby, the down counter is continuously loaded with the register contents. After start of calculation, the down counter becomes separated from the register and starts counting. The register can then be reloaded by the CPU for the next calculation cycle. As soon as the down counter reaches the state 00H, the calculation cycle is terminated.

**Limit registers**

The calculation unit has two 8-bit limit registers (SFRs) WRLIM and RDLIM. WRLIM controls the start of Ao result output to RAM, while RDLIM controls the length of Ai and Xi input operands. RDLIM is split into two 4-bit registers. RDLIM (7-4) carry the READ limit for the Xi operand and RDLIM (3-0) the READ limit for Ai.

The contents of WRLIM, RDLIM (7-4), RDLIM (3-0) are compared to the contents of the Cycle Counter CNTCYCL during calculation. As Xi and Ai limits are both only 4-bits wide, these values are expanded to 8-bits for comparison by adding four leading zeros.

**WRITE LIMIT REGISTER WRLIM:**

- as long as the CNTCYCL contents are greater than the WRLIM's contents, writing of the Ao output result to RAM is inhibited.

**READ LIMIT REGISTER RDLIM:**

- READ Xi limit (upper 4-bits): when the CNTCYCL contents has reached the Xi limit, the READ value of the Xi operand is 0.
- READ Ai limit (lower 4-bits): when CNTCYCL contents has reached the Ai limit, the READ value of the Ai operand is 0.

The limit registers can be read and written by the CPU. Because of their pipeline structure, they can be re-loaded whilst the calculation unit is active. Both registers are cleared to 00H during reset.

**CXOR register**

This is an 8-bit wide SFR that provides one operand for an exclusive-OR operation on the calculation result. It can be read and written by the CPU, the default value after reset is 0.

**Calculation unit interrupt**

At the end of a calculation cycle, an interrupt request is generated (DONE = 1). The DONE flag has to be reset by software during the interrupt service routine. The calculation unit's operation cannot be interrupted by any other interrupt. Interrupt requests are pending until the end of the calculation cycle. They will be acknowledged during the interrupt service routine.

**Parallel operation of CPU and calculation unit**

The performance of the calculation unit degrades if pointer and control register initializations are done in between two consecutive calculation cycles. The full calculation speed is reached when these initializations are carried out by the CPU in parallel to a computation

## Secured 8-bit microcontroller

## 83C852

cycle of the calculation unit (see Fig. 13). Thus with parallel initializations, when a current computation cycle is completed, all of the required initializations have already been done to enable the next computation cycle of the calculation unit to start immediately.

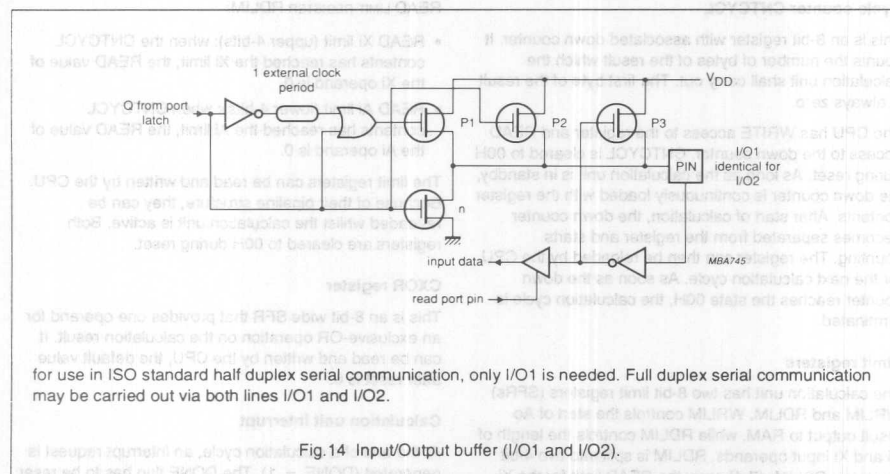
## IO register

The 83C852 has 2 I/O lines: I/O1 and I/O2. Line I/O1 is represented by IO1 (bit 0) and line I/O2 is represented by IO2 (bit 1) of the IO register (SFR). IO bits: 7, 6, 5, 4, 3 and 2 are don't care.

Table 12 I/O SFR

-	-	-	-	-	-	102	101
7	6	5	4	3	2	1	0

Either I/O line can be used independently from the other as an input or as an output. For an I/O line to be used as an input, a set 1 must first be written to its port-latch. See Fig. 14. The strong output driver FET P1 is turned off after one external clock period. The pin is then pulled HIGH by the weak pull-up FETs P2 and P3. It can be pulled LOW by an external source. After a hardware reset, both port latches contain a set 1 and both lines I/O1 and I/O2 are in Input mode.



## Secured 8-bit microcontroller

## 883C852

## Timers

The 83C852 has two 16-bit timer registers Timer 0 and Timer 1. The timer registers are incremented each machine cycle and are thus capable of counting machine cycles. Since a machine cycle consists of 6 external clock periods, the count rate is 1/6 of the clock frequency. Each timer has three operating modes:

- Mode 0 = 13-bit timer
- Mode 1 = 16-bit timer
- Mode 2 = 8-bit timer with auto-reload.

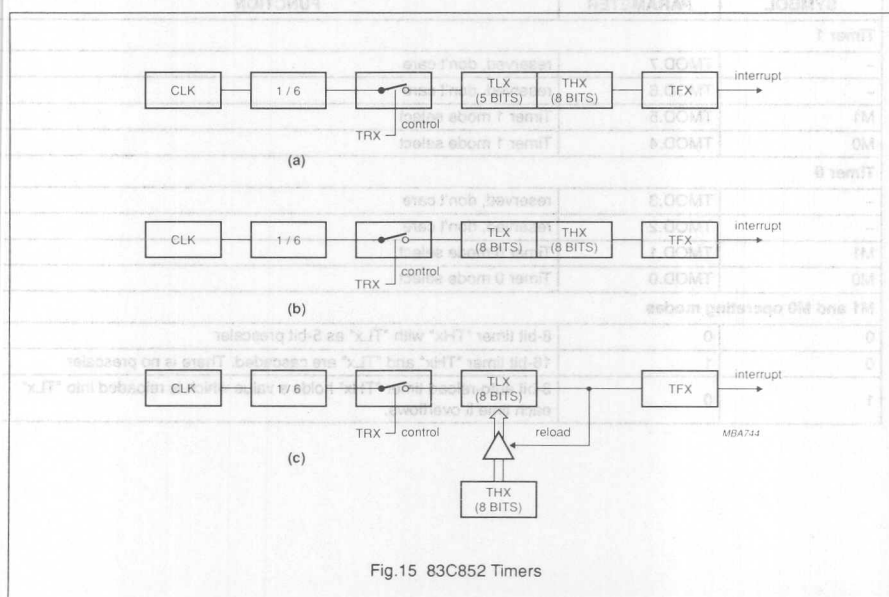


Fig.15 83C852 Timers

Secured 8-bit microcontroller

83C852

Timers 0 and 1 are controlled via the two SFRs: Timer Mode Control (TMOD) and Timer Control/External Interrupt Control (TCON).

Table 13 TMOD SFR

TIMER 1				TIMER 2			
—	—	M1	M0	—	—	M1	M0
7	6	5	4	3	2	1	0

Table 14 Description of the TMOD bits

SYMBOL	PARAMETER	FUNCTION
Timer 1		
—	TMOD.7	reserved, don't care
—	TMOD.6	reserved, don't care
M1	TMOD.5	Timer 1 mode select
M0	TMOD.4	Timer 1 mode select
Timer 0		
—	TMOD.3	reserved, don't care
—	TMOD.2	reserved, don't care
M1	TMOD.1	Timer 0 mode select
M0	TMOD.0	Timer 0 mode select
M1 and M0 operating modes		
0	0	8-bit timer "THx" with "TLx" as 5-bit prescaler
0	1	16-bit timer "THx" and "TLx" are cascaded. There is no prescaler
1	0	8-bit auto-reload timer "THx" holds a value which is reloaded into "TLx" each time it overflows.



## Secured 8-bit microcontroller

83C852

Table 15 TCON SFR

TIMER CONTROL				EXTERNAL INTERRUPT CONTROL			
TF1	TR1	TF0	TR0	—	IOSW	IE0	IT0
7	6	5	4	3	2	1	0

Table 16 Description of the TCON bits

SYMBOL	PARAMETER	FUNCTION
<b>Timer control</b>		
TF1	TCON.7	Timer 1 overflow flag. Set by hardware on Timer 1 overflow. Cleared by hardware when processor vectors to interrupt routine.
TR1	TCON.6	Timer 1 run control bit. Set/cleared by software to turn Timer 1 ON/OFF.
TF0	TCON.5	Timer 0 overflow flag. Set by hardware on Timer 0 overflow. Cleared by hardware when processor vectors to interrupt routine.
TR0	TCON.4	Timer 0 run control bit. Set/cleared by software to turn Timer 0 ON/OFF.
<b>External Interrupt Control</b>		
—	TCON.3	reserved, don't care.
IOSW	TCON.2	switch for external interrupt source: 0 = I/O1 is used as external interrupt source; 1 = I/O2 is used as external interrupt source.
IE0	TCON.1	is the external interrupt 0 edge flag. If IT0 is set 1, the IE0 bit is set 1 by hardware when the external interrupt source (either I/O1 pin or I/O2 pin) is detected to have made a 1 to 0 transition. The IE0 bit is cleared by hardware when the processor transfers control to the interrupt service routine.
IT0	TCON.0	determines whether external interrupt is edge-triggered or level-triggered. If IT0 is set 1: external interrupt 0 is edge-triggered. If IT0 is reset 0, external interrupt 0 is triggered by a detected LOW at the external interrupt source.

## Secured 8-bit microcontroller

## 83C852

## Interrupt system

The 83C852 has five interrupt sources, each can be programmed to one of two priority interrupt levels, either HIGH or LOW. The five interrupt sources are listed below:

1. I/O: external request from either I/O line
2. Timer 0: overflow from Timer 0
3. Cell: end of calculation
4. Timer 1: overflow from Timer 1
5. EEPROM: completion of EEPROM programming.

Each interrupt source can be individually enabled or disabled, all interrupt sources can also be globally enabled or disabled.

Each interrupt source can be programmed to either a HIGH or a LOW priority interrupt level. A LOW can be interrupted by a HIGH priority interrupt, but not by another LOW priority interrupt. A HIGH priority interrupt cannot be interrupted.

Only one of the I/O lines (either I/O1 or I/O2) can be used as an external interrupt source at a time. This selection is made by IOSW bit from TCON register.

## Interrupt vectors

The microcontroller acknowledges a request from an interrupt by a hardware subroutine call. It pushes the contents of the PC (program counter) into the stack, but it does not save the PSW (program status word). PC is reloaded with an address that depends on the source of the interrupt request, as shown below:

Address	Source
0003H	I/O1 or I/O2
000BH	Timer 0 Overflow
0013H	end of calculation
001BH	Timer 1 Overflow
0023H	completion of EEPROM programming.

## Interrupt registers IE and IP

## INTERRUPT ENABLE REGISTER IE

Each source can be individually enabled or disabled by setting or clearing the corresponding bit inside the SFR Interrupt Enable register IE. All interrupt sources can also be globally enabled or disabled.

Table 17. IE SFR

EA	EE	ET1	EC	ET0	EX0
7	6	5	4	3	2

Table 18. Description of the IE bits

SYMBOL	PARAMETER	FUNCTION
EA	IE.7	general enable/disable control 0 = no interrupt is enabled 1 = any individually enabled interrupt will be accepted
—	IE.6	reserved, don't care
—	IE.5	reserved, don't care
EE	IE.4	enable EEPROM interrupt
ET1	IE.3	enable Timer 1 interrupt
EC	IE.2	enable calculation unit interrupt
ET0	IE.1	enable Timer 0 interrupt
EX0	IE.0	enable external 0 interrupt (from I/O)

## Secured 8-bit microcontroller

83C852

Table 19 IP SFR

IP.7	IP.6	IP.5	PE	PT1	PCU	PT0	PX0
7	6	5	4	3	2	1	0

Table 20 Description of the IP bits

SYMBOL	PARAMETER	FUNCTION
—	IP.7	reserved, don't care
—	IP.6	reserved, don't care
—	IP.5	reserved, don't care
PE	IP.4	EEPROM interrupt level
PT1	IP.3	Timer 1 interrupt level
PCU	IP.2	calculation unit interrupt level
PT0	IP.1	Timer 0 interrupt level
PX0	IP.0	external 0 interrupt level

## INTERRUPT PRIORITY REGISTER IP

The interrupt level is selected within the SFR Interrupt Priority register IP. Setting a bit to '1' selects HIGH priority.

## Hardware security

## OPERATING MODE

The microcontroller has two operating modes:

- User mode
- Test mode

The test mode is permanently disabled once the test has been performed.

## Low frequency sensor

The low frequency detector circuit triggers a reset of the CPU when the clock frequency falls below  $f_{CLK}$  minimum (approximately 500 kHz). When the clock frequency rises above  $f_{CLK}$  minimum the reset is de-activated.

## Power ON/OFF reset

The power ON/OFF reset circuit triggers a reset of the CPU when the power supply falls below  $V_{DD}$  minimum (approximately 3.5 V). When the power supply rises above  $V_{DD}$  minimum, this reset is de-activated. When the power-down mode is active (PD is set 1) the power ON/OFF reset is de-activated.

## Idle mode and power-down mode

## IDLE MODE

The 83C852 provides two power saving operational modes, the idle mode and the power-down mode. In the idle mode, the CPU enters a sleep routine whilst some of the on-chip peripherals (timers and interrupt system) remain active. The contents of the RAM and SFRs remain unchanged during the duration of an idle mode. The idle mode can be terminated by an enabled interrupt or by a hardware reset. Besides stopping the CPU, the idle mode terminates EEPROM write operations and stops operation of the calculation unit.

## POWER-DOWN MODE

In the power-down mode, all on-chip internal clocks are frozen. The CPU and all on-chip peripherals stop working. The only exit from a power-down mode is by a hardware reset. The on-chip RAM and SFRs retain their values until the power-down mode is terminated. Reset redefines the SFRs, but does not change the RAM contents. The  $V_{DD}$  supply can be reduced to 2 V whilst the power-down mode is active. Both modes are activated by software via the SFR Power Control register PCON. PCON is not bit addressable.

Secured 8-bit microcontroller

83C852

Table 21 PCON SFR

7	6	5	4	3	2	1	0
—	—	—	—	GF1	GF0	PD	IDL

Table 22 Description of the PCON bits

SYMBOL	PARAMETER	FUNCTION
—	PCON.7	reserved, don't care
—	PCON.6	reserved, don't care
—	PCON.5	reserved, don't care
—	PCON.4	reserved, don't care
GF1	PCON.3	general purpose flag bit
GF0	PCON.2	general purpose flag bit
PD	PCON.1	enter power-down mode when set; note 1
IDL	PCON.0	enter idle mode when set; note 1

Note

1. If a logic 1 is written to PD and IDL at the same time, PD takes precedence.

## Secured 8-bit microcontroller

83C852

## SFRs memory mapping

Table 23

The 83C852 has the following 33 Special Function Registers (SFRs) available to the user.

SFRs ADDRESS	SYMBOL	RESET VALUE	FUNCTION
FBH	WRLIM	0000.0000B	WRITE limit register for calculation unit
FAH	RDLIM	0000.0000B	READ limit for calculation unit
F9H	CNTCYCL	0000.0000B	cycle counter for calculation unit
F7H	ECNTRL2	XXXX.0000B	EEPROM control register (test modes)
F6H	ECNTRL1	0000.0000B	EEPROM control register (user modes)
F5H	ETIM	0000.1000B	EEPROM timer register
F4H	EDAT	XXXX.XXXXB	EEPROM data register
F3H	EADRH	1000.0000B	EEPROM address register HIGH
F2H	EADRL1	0000.0000B	EEPROM address register 1 LOW, address pointer AIPE for calculation unit
F1H	EADRL2	0000.0000B	EEPROM address register 2 LOW, address pointer XIPE for calculation unit
F0H	B	0000.0000B	B register
E0H	ACC	0000.0000B	accumulator
D0H	PSW	0000.0000B	program status word
B8H	IP	XXX0.0000B	interrupt priority register
B0H	IO	XXXX.XX11B	I/O register
A8H	IE	0XX0.0000B	interrupt enable register
A7H	AOPR	0000.0000B	AOPR register for calculation unit
A6H	APR	0000.0000B	APR register for calculation unit
A5H	XIPR	0000.0000B	XIPR register for calculation unit
A4H	AIPR	0000.0000B	AIPR register for calculation unit
A3H	CXOR	0000.0000B	CXOR register for calculation unit
99H	CMD	0000.0000B	command register for calculation unit
98H	CMDSTAT	XXX0.0000B	command and status register for calculation unit
8DH	TH1	0000.0000B	Timer 1 HIGH
8CH	TH0	0000.0000B	Timer 0 HIGH
8BH	TL1	0000.0000B	Timer 1 LOW
8AH	TL0	0000.0000B	Timer 0 LOW
89H	TMOD	XX00.XX00B	Timer 0 and 1 mode control
88H	TCON	0000.X000B	Timer 0 and 1 control and external interrupt control
87H	PCON	XXXX.0000B	power control register
83H	DPH	0000.0000B	data pointer HIGH
82H	DPL	0000.0000B	data pointer LOW
81H	SP	0000.0111B	stack pointer



## 830C852

Fig.16 SFRs mapping.

## Secured 8-bit microcontroller

83C852

## INSTRUCTION SET

The instruction set consists of 49 single-byte, 46 two-byte and 16 three-byte instructions. When using a 6 MHz external clock, 64 instructions execute in 1 cycle (1  $\mu$ s) and 45 instructions execute in 2 cycles (2  $\mu$ s). Multiply and divide instructions execute in 4 cycles (4  $\mu$ s).

## INSTRUCTION SET DESCRIPTION

For data addressing modes, Hexadecimal opcode cross-reference and invalid instructions, see Table notes.

OPCODE (HEX)	CYCLES	BYTES	DESCRIPTION	MNEMONIC
<b>Arithmetic operation</b>				
2*	1	1	Add register to A	ADD A,Rr
25	1	2	Add direct byte to A	ADD A,direct
26, 27	1	1	Add indirect RAM to A	ADD A,@Ri
24	1	2	Add immediate data to A	ADD A,#data
3*	1	1	Add register to A with carry flag	ADDC A,Rr
35	1	2	Add direct byte to A with carry flag	ADDC A,direct
36, 37	1	1	Add indirect RAM to A with carry flag	ADDC A,@Ri
34	1	2	Add immediate data to A with carry flag	ADDC A,#data
9*	1	1	Subtract register from A with borrow	SUBB A,Rr
95	1	2	Subtract direct byte from A with borrow	SUBB A,direct
96, 97	1	1	Subtract indirect RAM from A with borrow	SUBB A,@Ri
94	1	2	Subtract immediate data from A with borrow	SUBB A,#data
04	1	1	Increment A	INC A
0*	1	1	Increment register	INC Rr
05	1	2	Increment direct byte	INC direct
06, 07	1	1	Increment indirect RAM	INC @Ri
14	1	1	Decrement A	DEC A
1*	1	1	Decrement register	DEC Rr
15	1	2	Decrement direct byte	DEC direct
16, 17	1	1	Decrement indirect RAM	DEC @Ri
A3	2	1	Increment data pointer	INC DPTR
A4	4	1	Multiply A & B	MUL AB
84	4	1	Divide A by B	DIV AB
D4	1	1	Decimal adjust A	DA A

## Secured 8-bit microcontroller

83C852

Instruction set description (continued)

MNEMONIC	DESCRIPTION	BYTES	CYCLES	OPCODE (HEX)
<b>Logic operations</b>				
ANL A,Rr	AND register to A	1	1	5*
ANL A,direct	AND direct byte to A	2	1	55
ANL A,@Ri	AND indirect RAM to A	1	1	56, 57
ANL A,#data	AND immediate data to A	2	1	54
ANL direct,A	AND A to direct byte	2	1	52
ANL direct,#data	AND immediate data to direct byte	3	2	53
ORL A,Rr	OR register to A	1	1	4*
ORL A,direct	OR direct byte to A	2	1	45
ORL A,@Ri	OR indirect RAM to A	1	1	46, 47
ORL A,#data	OR immediate data to A	2	1	44
ORL direct,A	OR A to direct byte	2	1	42
ORL direct,#data	OR immediate data to direct byte	3	2	43
XRL A,Rr	Exclusive-OR register to A	1	1	6*
XRL A,direct	Exclusive-OR direct byte to A	2	1	65
XRL A,@Ri	Exclusive-OR indirect RAM to A	1	1	66, 67
XRL A,#data	Exclusive-OR immediate data to A	2	1	64
XRL direct,A	Exclusive-OR A to direct byte	2	1	62
XRL direct,#data	Exclusive-OR immediate data to direct byte	3	2	63
CLR A	Clear A	1	1	E4
CPL A	Complement A	1	1	F4
RL A	Rotate A left	1	1	23
RLC A	Rotate A left through the carry flag	1	1	33
RR A	Rotate A right	1	1	03
RRC A	Rotate A right through the carry flag	1	1	13
SWAP A	Swap nibbles within A	1	1	C4

## Secured 8-bit microcontroller

## 83C852

## Instruction set description (continued)

MNEMONIC	BYTES	DESCRIPTION	BYTES	CYCLES	OPCODE (HEX)
<b>Data transfer</b>					
MOV A,Rr	1	Move register to A	1	1	E*
MOV A,direct**	2	Move direct byte to A	2	1	E5
MOV A,@Ri	1	Move indirect RAM to A	1	1	E6, E7
MOV A,#data	2	Move immediate data to A	2	1	74
MOV Rr,A	1	Move A to register	1	1	F*
MOV Rr,direct	2	Move direct byte to register	2	2	A*
MOV Rr,#data	2	Move immediate data to register	2	1	7*
MOV direct,A	2	Move A to direct byte	2	1	F5
MOV direct,Rr	2	Move register to direct byte	2	2	8*
MOV direct,direct	3	Move direct byte to direct	3	2	85
MOV direct,@Ri	2	Move indirect RAM to direct byte	2	2	86, 87
MOV direct,#data	3	Move immediate data to direct byte	3	2	75
MOV @Ri,A	1	Move A to indirect RAM	1	1	F6, F7
MOV @Ri,direct	2	Move direct byte to indirect RAM	2	2	A6, A7
MOV @Ri,#data	2	Move immediate data to indirect RAM	2	1	76, 77
MOV DPTR,#data 16	3	Load data pointer with a 16-bit constant	3	2	90
MOVC A,@A+DPTR	1	Move code byte relative to DPTR to A	1	2	93
MOVC A,@A+PC	1	Move code byte relative to PC to A	1	2	83
PUSH direct	2	Push direct byte onto stack	2	2	C0
POP direct	2	Pop direct byte from stack	2	2	D0
XCH A,Rr	1	Exchange register with A	1	1	C*
XCH A,direct	2	Exchange direct byte with A	2	1	C5
XCH A,@Ri	1	Exchange indirect RAM with A	1	1	C6, C7
XCHD A,@Ri	1	Exchange LOW-order digit indirect RAM with A	1	1	D6, D7
<b>Note: the following MOVX instructions of 80C51 set are not applicable to 83C852</b>					
MOVX A,@Ri	1	Move external RAM (8-bit address) to A	1	2	E2, E3
MOVX A,DPTR	1	Move external RAM (16-bit address) to A	1	2	E0
MOVX @Ri,A	1	Move A to external RAM (8-bit address)	1	2	F2, F3
MOVX @DPTR,A	1	Move A to external RAM (16-bit address)	1	2	F0

## Secured 8-bit microcontroller

83C852

## Instruction set description (continued)

OPCODE (HEX)	MNEMONIC	DESCRIPTION	BYTES	CYCLES	OPCODE (HEX)
<b>Boolean variable manipulation</b>					
CLR C	C	Clear carry flag	1	1	C3
CLR bit	bit	Clear direct bit	2	1	C2
SETB C	C	Set carry flag	1	1	D3
SETB bit	bit	Set direct bit	2	1	D2
CPL C	C	Complement carry flag	1	1	B3
CPL bit	bit	Complement direct bit	2	1	B2
ANL C,bit	C,bit	AND direct bit to carry flag	2	2	82
ANL C,/bit	C,/bit	AND complement of direct bit to carry flag	2	2	B0
ORL C,bit	C,bit	OR direct bit to carry flag	2	2	72
ORL C,/bit	C,/bit	OR complement of direct bit to carry flag	2	2	A0
MOV C,bit	C,bit	Move direct bit to carry flag	2	1	A2
MOV bit,C	bit,C	Move carry flag to direct bit	2	2	92
<b>Program and machine control</b>					
ACALL addr11	addr11	Absolute subroutine call	2	2	•1addr
LCALL addr16	addr16	Long subroutine call	3	2	12
RET		Return from subroutine	1	2	22
RETI		Return from interrupt	1	2	32
AJMP addr11	addr11	Absolute jump	2	2	•1addr
LJMP addr16	addr16	Long jump	3	2	02
SJMP rel	rel	Short jump (relative address)	2	2	80
JMP @A+DPTR	@A+DPTR	Jump indirect relative to the DPTR	1	2	73
JZ rel	rel	Jump if A is zero	2	2	60
JNZ rel	rel	Jump if A is not zero	2	2	70
JC rel	rel	Jump if carry flag is set	2	2	40
JNC rel	rel	Jump if carry flag is not set	2	2	50
JB bit,rel	bit,rel	Jump if direct bit is set	3	2	20
JNB bit,rel	bit,rel	Jump if direct bit is not set	3	2	30
JBC bit,rel	bit,rel	Jump if direct bit is set and clear bit	3	2	10
CJNE A,direct,rel	A,direct,rel	Compare direct to A and jump if not equal	3	2	B5
CJNE A,#data,rel	A,#data,rel	Compare immediate to A and jump if not equal	3	2	B4
CJNE Rr,#data,rel	Rr,#data,rel	Compare immed. to reg. and jump if not equal	3	2	B*
CJNE @Ri,#data,rel	@Ri,#data,rel	Compare immed. to ind. and jump if not equal	3	2	B6, B7
DJNZ Rr,rel	Rr,rel	Decrement register and jump if not zero	2	2	D*
DJNZ direct,rel	direct,rel	Decrement direct and jump if not zero	3	2	D5
NOP		No operation	1	1	00



## Secured 8-bit microcontroller

80C852

## NOTES TO INSTRUCTION SET TABLE

mnemonic	DESCRIPTION
<b>Data addressing modes</b>	
Rn	working register R0-R7.
direct	128 internal RAM locations and any special function register (SFR).
@Ri	indirect internal RAM location addressed by register R0 or R1 of the actual register bank.
#data	8-bit constant included in instruction.
#data 16	16-bit constant included as bytes 2 and 3 of instruction.
bit	direct addressed bit in internal RAM or SFR.
addr16	16-bit destination address. Used by LCALL and LJMP. The branch will be anywhere within the 64K byte program memory address space.
addr11	11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2K byte page of program memory as the first byte of the following instruction.
rel	Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is -128 to +127 bytes relative to first byte of the following instruction.
<b>Hexadecimal opcode cross-reference</b>	
•	8, 9, A, B, C, D, E, F.
•	11, 31, 51, 71, 91, B1, D1, F1.
•	01, 21, 41, 61, 81, A1, C1, E1.
<b>Invalid instructions:</b>	
note **	MOV A, ACC is not a valid instruction.
MOVX	MOVX instructions of 80C51 set are not applicable to 83C852.

## Secured 8-bit microcontroller

83C852

## INSTRUCTION MAP

MOVX instructions not applicable

	first hexadecimal character of opcode				second hexadecimal character of opcode											
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	AJMP addr11	LJMP addr16	RR A	INC A	INC dir	INC @Ri 0	1	INC Rr 0	1	2	3	4	5	6	7
1	JBC bit,rel	ACALL addr11	LCALL addr16	RRC A	DEC A	DEC dir	DEC @Ri 0	1	DEC Rr 0	1	2	3	4	5	6	7
2	JB bit,rel	AJMP addr11	RET	RL A	ADD A,#data	ADD A,dir	ADD A,@Ri 0	1	ADD A,Rr 0	1	2	3	4	5	6	7
3	JNB bit,rel	ACALL addr11	RETI	RLC A	ADDC A,#data	ADDC A,dir	ADDC A,@Ri 0	1	ADDC A,Rr 0	1	2	3	4	5	6	7
4	JC rel	AJMP addr11	ORL dir,A	ORL dir,#data	ORL A,#data	ORL A,dir	ORL A,@Ri 0	1	ORL A,Rr 0	1	2	3	4	5	6	7
5	JNC rel	ACALL addr11	ANL dir,A	ANL dir,#data	ANL A,#data	ANL A,dir	ANL A,@Ri 0	1	ANL A,Rr 0	1	2	3	4	5	6	7
6	JZ rel	AJMP addr11	XRL dir,A	XRL dir,#data	XRL A,#data	XRL A,dir	XRL A,@Ri 0	1	XRL A,Rr 0	1	2	3	4	5	6	7
7	JNZ rel	ACALL addr11	ORL C,bit	JMP @A+DPTR	MOV A,#data	MOV dir,#data	MOV @Ri,#data 0	1	MOV Rr,#data 0	1	2	3	4	5	6	7
8	SJMP rel	AJMP addr11	ANL C,bit	MOVC A,@A+PC	DIV AB	MOV dir,dir	MOV dir,@Ri 0	1	MOV dir,Rr 0	1	2	3	4	5	6	7
9	MOV DPTR, #data	ACALL addr11	MOV bit,C	MOVC A,@A+DPTR	SUBB A,#data	SUBB A,dir	SUBB A,@Ri 0	1	SUBB A,Rr 0	1	2	3	4	5	6	7
A	ORL C,/bit	AJMP addr11	MOV C,bit	INC DPTR	MUL AB		MOV @Ri,dir 0	1	MOV Rr,dir 0	1	2	3	4	5	6	7
B	ANL C,/bit	ACALL addr11	CPL bit	CPL C	CJNE A, #data,rel	CJNE A,dir,rel	CJNE @Ri,#data,rel 0	1	CJNE Rr,#data,rel 0	1	2	3	4	5	6	7
C	PUSH dir	AJMP addr11	CLR bit	CLR C	SWAP A	XCH A,dir	XCH A,@Ri 0	1	XCH A,Rr 0	1	2	3	4	5	6	7
D	POP dir	ACALL addr11	SETB bit	SETB C	DA A	DJNZ dir,rel	XCHD A,@Ri 0	1	DJNZ Rr,rel 0	1	2	3	4	5	6	7
E	MOVX A,@DPTR	AJMP addr11	MOVX A,@Ri 0	1	CLR A	MOV A,dir	MOV A,@Ri 0	1	MOV A,Rr 0	1	2	3	4	5	6	7
F	MOVX @DPTR,A	ACALL addr11	MOVX @Ri,A 0	1	CPL A	MOV dir,A	MOV @Ri,A 0	1	MOV Rr,A 0	1	2	3	4	5	6	7

\* MOV A,ACC is not a valid instruction.

## Secured 8-bit microcontroller

83C852

## ISO INFORMATION

The following ISO characteristics information as applicable to this data sheet may be superseded. Please ensure that the latest version of ISO information is studied for relevant features.

## ISO ELECTRICAL CHARACTERISTICS

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
<b>I/O</b>						
$V_{IH}$	input voltage HIGH	$I_{IH (max)} = \pm 20 \mu A$	$0.7 \times V_{DD}$	—	$V_{DD} + 0.3$	V
$V_{IL}$	input voltage LOW	$I_{IL (max)} = -1 \text{ mA}$	-0.3	—	0.8	V
$V_{OH}$	output voltage HIGH	$I_{OH (max)} = -20 \mu A$ ; note 1	3.8	—	$V_{DD}$	V
$V_{OL}$	output voltage LOW	$I_{OL (max)} = +1 \text{ mA}$	0	—	0.4	V
$C_{IO}$	input/output pin capacitance		—	—	30	pF
$t_{r/f}$	I/O rise/fall times	$C_{IN} = 30 \text{ pF}$ ; $C_{OUT} = 30 \text{ pF}$	—	—	1	$\mu s$
<b>CLK</b>						
$V_{IH}$	input voltage HIGH	$I_{IH (max)} = \pm 20 \mu A$	$0.7 \times V_{DD}$	—	$V_{DD} + 0.3$	V
$V_{IL}$	input voltage LOW	$I_{IL (max)} = \pm 200 \mu A$	-0.3	—	0.5	V
$C_I$	input pin capacitance		—	—	30	pF
$t_{r/f}$	CLK rise/fall times	$C_{IN} = 30 \text{ pF}$	—	—	9% of period with a max. of 0.5 $\mu s$	$\mu s$
<b>RESET</b>						
$V_{IH}$	input voltage HIGH	$I_{IH (max)} = \pm 20 \mu A$	$0.7 \times V_{DD}$	—	$V_{DD} + 0.3$	V
$V_{IL}$	input voltage LOW	$I_{IL (max)} = \pm 200 \mu A$	-0.3	—	0.5	V

## Note

1. It is assumed that a pull-up resistor is used in the interface device (recommend value = 20 k $\Omega$ ).

## LIMITING VALUES

In accordance with the Absolute Maximum System (IEC 134)

SYMBOL	PARAMETER	MIN.	MAX.	UNIT
$V_i$	input voltage on any pin with respect to ground ( $V_{SS}$ )	-0.5	$\pm 6.5$	V
$I_i, I_o$	input/output current on I/O1 or I/O2 pin	—	$\pm 5$	mA
$P_{tot}$	total power dissipation per package	—	1	W
$T_{stg}$	storage temperature range	-65	150	$^{\circ}C$
$T_{amb}$	operating ambient temperature range	0	70	$^{\circ}C$

## HANDLING

Inputs and outputs are protected against electrostatic discharge in normal handling. However, to be totally safe, it is desirable to take precautions appropriate to handling MOS devices (see 'Handling MOS Devices').

## Secured 8-bit microcontroller

83C852

## CHARACTERISTICS

 $V_{DD} = 5\text{ V}$  ( $\pm 10\%$ );  $V_{SS} = 0\text{ V}$ ;  $T_{amb} = 0$  to  $70\text{ }^{\circ}\text{C}$ ; all voltages with respect to  $V_{SS}$  unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
<b>DC</b>						
$V_{DD}$	supply voltage range		4.5	—	5.5	V
$I_{DD}$	supply current	$f_{CLK} = 3.57\text{ MHz}$	—	—	10	mA
$I_{DD}$	supply current operating mode	$f_{CLK} = 6.0\text{ MHz}$	—	—	15	mA
$I_{ID}$	supply current idle mode	$f_{CLK} = 6.0\text{ MHz}$	—	—	3	mA
$I_{PD}$	power-down current	$2\text{ V} \leq V_{PD} \leq V_{DD}\text{ max.}$	—	—	100	$\mu\text{A}$
<b>I/O1; I/O2</b>						
$V_{IL}$	input voltage LOW		-0.5	—	$0.2 V_{DD}$	V
$V_{IH}$	input voltage HIGH		$0.2 V_{DD}$	—	$V_{DD} + 0.5$	V
$I_{IL}$	input current LOW	$V_I = +0.45\text{ V}$	—	—	-50	$\mu\text{A}$
$I_{IH}$	input current HIGH-to-LOW		—	—	650	$\mu\text{A}$
$V_{OH}$	output voltage HIGH	$I_{OH} = -20\text{ }\mu\text{A}$	3.8	—	—	V
$V_{OL}$	output voltage LOW	$I_{OL} = 1.0\text{ mA}$	—	—	0.4	V
$V_{IL}$	RESET; CLK input voltage LOW		-0.5	—	$0.2 V_{DD} - 0.1$	V
$V_{IH}$	RESET; CLK input voltage HIGH		$0.7 V_{DD}$	—	$V_{DD} + 0.5$	V
$I_U$	input leakage current (RESET; CLK)	$0.45 < V_I < V_{DD}$	—	—	$\pm 10$	$\mu\text{A}$
ESD	ESD protection	$C = 100\text{ pF}$ ; $R = 1.5\text{ k}$	—	—	2.0	kV
<b>AC</b>						
$f_{CLK}$	external clock frequency	internal operating frequency = $f_{CLK}$	1	—	6	MHz
$t_{CYC}$	cycle time		1	—	—	$\mu\text{s}$
$t_{CLK}$	clock pulse width		45	—	55	%
$t_r$	clock rise time		—	—	tb <sub>f</sub>	ns
$t_f$	clock fall time		—	—	tb <sub>f</sub>	ns
$t_{POR}$	power-on reset delay		tb <sub>f</sub>	—	tb <sub>f</sub>	tb <sub>f</sub>
$t_{rw}$	reset pulse width		$12/f_{CLK}$	—	—	s
$t_e$	EEPROM ERASE time		—	5.0	—	ms
$t_w$	EEPROM WRITE time		—	5.0	—	ms
$t_s$	EEPROM data retention time	$T_{amb} = 55\text{ }^{\circ}\text{C}$	10.0	—	—	hrs
$N_{erw}$	EEPROM endurance (number of erase/write cycles)	$t_e = 5\text{ ms}$ ; $t_w = 5\text{ ms}$	10 000	—	—	cycles
C	I/O1; I/O2; RESET; $f_{CLK}$ pin capacitance	$f_{CLK} = 1\text{ MHz}$ ; $T_{amb} = 25\text{ }^{\circ}\text{C}$	—	—	10	pF

# Section 4

## Inter-Integrated (I<sup>2</sup>C) Circuit Bus

80C51-Based  
8-Bit Microcontrollers

CONTENTS

The I <sup>2</sup> C-bus and how to use it (including specification) .....	1141
I <sup>2</sup> C-bus peripheral selection guide .....	1160
<b>82B715</b> I <sup>2</sup> C bus extender .....	1162





## The I<sup>2</sup>C-bus and how to use it

# The I<sup>2</sup>C-bus and how to use it (including specification)

### 1.0 THE I<sup>2</sup>C-BUS BENEFITS DESIGNERS AND MANUFACTURERS

In consumer electronics, telecommunications and industrial electronics, there are often many similarities between seemingly unrelated designs. For example, nearly every system includes:

- Some intelligent control, usually a single-chip microcontroller
- General purpose circuits like LCD drivers, remote I/O ports, RAM, EEPROM, or data converters
- Application-oriented circuits such as digital tuning and signal processing circuits for radio and video systems, or DTMF generators for telephones with tone dialling

To exploit these similarities to the benefit of both systems designers and equipment manufacturers, as well as to maximize hardware efficiency and circuit simplicity, Philips developed a simple bidirectional 2-wire bus for efficient inter-IC control. This bus is called the Inter IC or I<sup>2</sup>C-bus. At present, Philips' IC range includes more than 150 CMOS and bipolar I<sup>2</sup>C-bus compatible types for performing functions in all three of the previously mentioned categories. All I<sup>2</sup>C-bus compatible devices incorporate an on-chip interface which allows them to communicate directly with each other via the I<sup>2</sup>C-bus. This design concept solves the many interfacing problems encountered when designing digital control circuits.

Here are some of the features of the I<sup>2</sup>C-bus:

- Only two bus lines are required; a serial data line (SDA) and a serial clock line (SCL)
- Each device connected to the bus is software addressable by a unique address and simple master/slave relationships exist at all times; masters can operate as master-transmitters or as master-receivers
- It's a true multi-master bus including collision detection and arbitration to prevent data corruption if two or more masters simultaneously initiate data transfer
- Serial, 8-bit oriented, bidirectional data transfer: can be made at up to 100 kbit/s in the standard mode or up to 400 kbit/s in the fast mode
- On-chip filtering rejects spikes on the bus

data line to preserve data integrity

- The number of ICs that can be connected to the same bus is limited only by a maximum bus capacitance of 400 pF

Figure 1 shows two examples of I<sup>2</sup>C-bus applications.

### 1.1 Designer Benefits

I<sup>2</sup>C-bus compatible ICs allow a system design to rapidly progress directly from a functional block diagram to a prototype. Moreover, since they 'clip' directly onto the I<sup>2</sup>C-bus without any additional external interfacing, they allow a prototype system to be modified or upgraded simply by 'clipping' or 'unclipping' ICs to or from the bus.

Here are some of the features of I<sup>2</sup>C-bus compatible ICs which are particularly attractive to designers:

- Functional blocks on the block diagram correspond with the actual ICs; designs proceed rapidly from block diagram to final schematic
- No need to design bus interfaces because the I<sup>2</sup>C-bus interface is already integrated on-chip
- Integrated addressing and data-transfer protocol allow systems to be completely software-defined
- The same IC types can often be used in many different applications
- Design-time reduces as designers quickly become familiar with the frequently used functional blocks represented by I<sup>2</sup>C-bus compatible ICs
- ICs can be added to or removed from a system without affecting any other circuits on the bus
- Fault diagnosis and debugging are simple; malfunctions can be immediately traced
- Software development time can be reduced by assembling a library of reusable software modules.

In addition to these advantages, the CMOS ICs in the I<sup>2</sup>C-bus compatible range offer designers special features which are particularly attractive for portable equipment and battery-backed systems. They all have:

- Extremely low current consumption
- High noise immunity

- Wide supply voltage range
- Wide operating temperature range.

### 1.2 Manufacturer benefits

I<sup>2</sup>C-bus compatible ICs don't only assist designers, they also give a wide range of benefits to equipment manufacturers because:

- The simple 2-wire serial I<sup>2</sup>C-bus minimizes interconnections so ICs have fewer pins and there are not so many PCB tracks; result—smaller and less expensive PCBs
- The completely integrated I<sup>2</sup>C-bus protocol eliminates the need for address decoders and other 'glue logic'
- The multi-master capability of the I<sup>2</sup>C-bus allows rapid testing and alignment of end-user equipment via external connections to an assembly-line computer
- The availability of I<sup>2</sup>C-bus compatible ICs in SO (small outline), VSO (very small outline) as well as DIL packages reduces space requirements even more.

These are just some of the benefits. In addition, I<sup>2</sup>C-bus compatible ICs increase system design flexibility by allowing simple construction of equipment variants and easy upgrading to keep designs up-to-date. In this way, an entire family of equipment can be developed around a basic model. Upgrades for new equipment, or enhanced-feature models (i.e. extended memory, remote control, etc.) can then be produced simply by clipping the appropriate ICs onto the bus. If a larger ROM is needed, it's simply a matter of selecting a microcontroller with a larger ROM from our comprehensive range. As new ICs supersede older ones, it's easy to add new features to equipment or to increase its performance by simply unclipping the outdated IC from the bus and clipping on its successor.

### 1.3 The ACCESS.bus

Another attractive feature of the I<sup>2</sup>C-bus for designers and manufacturers is that its simple 2-wire nature and capability of software addressing make it an ideal platform for the ACCESS.bus (Fig.2). This is a lower-cost alternative for an RS-232C interface for connecting peripherals to a host computer via a simple 4-pin connector (see Section 19).

# The I<sup>2</sup>C-bus and how to use it

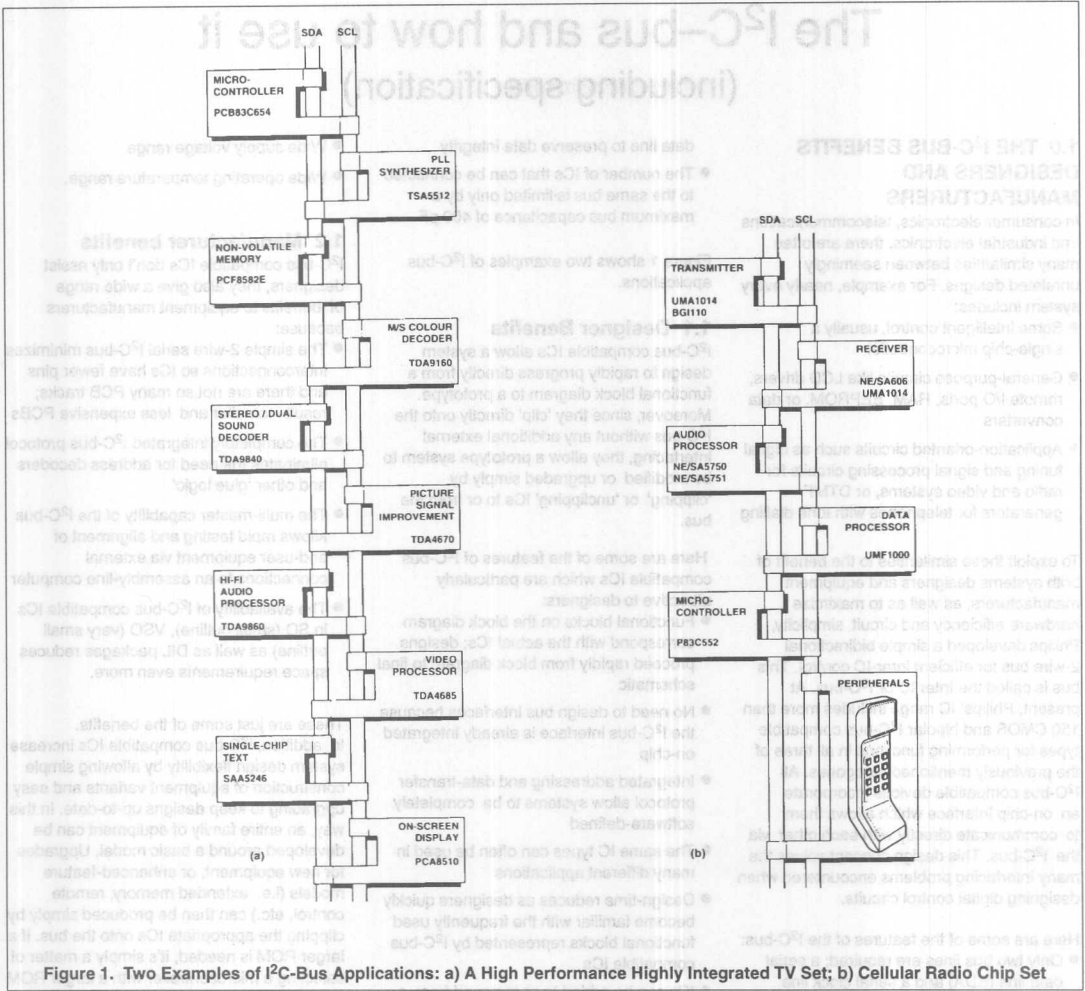


Figure 1. Two Examples of I<sup>2</sup>C-Bus Applications: a) A High Performance Highly Integrated TV Set; b) Cellular Radio Chip Set

Table 1. Definition of I<sup>2</sup>C-Bus Terminology

Term	Description
Transmitter	The device which sends the data to the bus
Receiver	The device which receives the data from the bus
Master	The device which initiates a transfer, generates clock signals and terminates a transfer
Slave	The device addressed by a master
Multi-master	More than one master can attempt to control the bus at the same time without corrupting the message
Arbitration	Procedure to ensure that, if more than one master simultaneously tries to control the bus, only one is allowed to do so and the message is not corrupted
Synchronization	Procedure to synchronize the clock signals of two or more devices

## The I<sup>2</sup>C-bus and how to use it

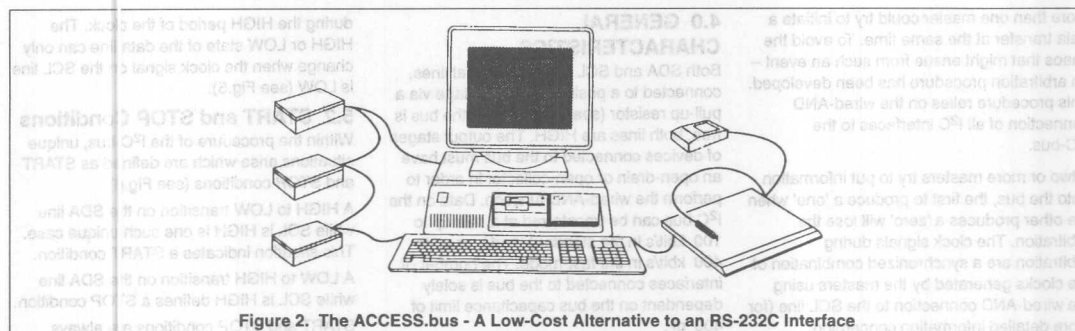


Figure 2. The ACCESS.bus - A Low-Cost Alternative to an RS-232C Interface

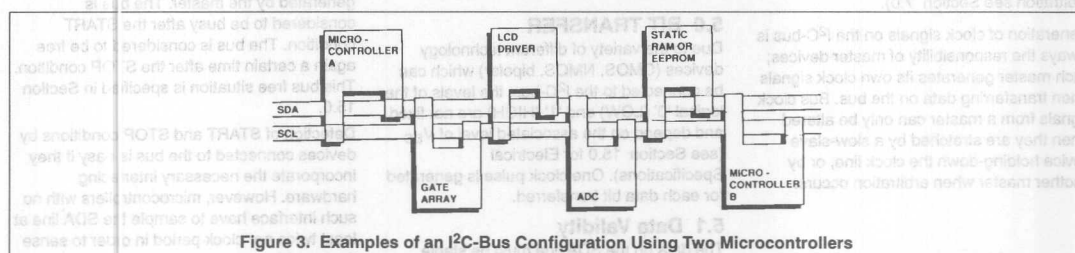


Figure 3. Examples of an I<sup>2</sup>C-Bus Configuration Using Two Microcontrollers

### 2.0 INTRODUCTION TO THE I<sup>2</sup>C-BUS SPECIFICATION

For 8-bit digital control applications, such as those requiring microcontrollers, certain design criteria can be established:

- A complete system usually consists of at least one microcontroller and other peripheral devices such as memories and I/O expanders
- The cost of connecting the various devices within the system must be minimized
- A system that performs a control function doesn't require high-speed data transfer
- Overall efficiency depends on the devices chosen and the nature of the interconnecting bus structure.

In order to produce a system to satisfy these criteria, a serial bus structure is needed. Although serial buses don't have the throughput capability of parallel buses, they do require less wiring and fewer IC connecting pins. However, a bus is not merely an interconnecting wire, it embodies all the formats and procedures for communication within the system.

Devices communicating with each other on a serial bus must have some form of protocol which avoids all possibilities of confusion, data loss and package of information. Fast devices must be able to communicate with slow devices. The system must not be dependent on the devices connected to it,

otherwise modifications or improvements would be impossible. A procedure has also to be devised to decide which device will be in control of the bus and when. And, if different devices with different clock speeds are connected to the bus, the bus clock source must be defined. All these criteria are involved in the specification of the I<sup>2</sup>C-bus.

### 3.0 THE I<sup>2</sup>C-BUS CONCEPT

The I<sup>2</sup>C-bus supports any IC fabrication process (NMOS, CMOS, bipolar). Two wires, serial data (SDA) and serial clock (SCL), carry information between the devices connected to the bus. Each device is recognised by a unique address – whether it's a microcontroller, LCD driver, memory or keyboard interface – and can operate as either a transmitter or receiver, depending on the function of the device. Obviously an LCD driver is only a receiver, whereas a memory can both receive and transmit data. In addition to transmitters and receivers, devices can also be considered as masters or slaves when performing data transfers (see Table 1). A master is the device which initiates a data transfer on the bus and generates the clock signals to permit that transfer. At that time, any device addressed is considered a slave.

The I<sup>2</sup>C-bus is a multi-master bus. This means that more than one device capable of

controlling the bus can be connected to it. As masters are usually micro-controllers, let's consider the case of a data transfer between two microcontrollers connected to the I<sup>2</sup>C-bus (Fig.3). This highlights the master-slave and receiver-transmitter relationships to be found on the I<sup>2</sup>C-bus. It should be noted that these relationships are

not permanent, but only depend on the direction of data transfer at that time. The transfer of data would proceed as follows:

1. Suppose microcontroller A wants to send information to microcontroller B:
  - microcontroller A (master), addresses microcontroller B (slave)
  - microcontroller A (master-transmitter), sends data to microcontroller B (slave-receiver)
  - microcontroller A terminates the transfer.
2. If microcontroller A wants to receive information from microcontroller B:
  - microcontroller A (master) addresses microcontroller B (slave)
  - microcontroller A (master-receiver) receives data from microcontroller B (slave-transmitter)
  - microcontroller A terminates the transfer.

Even in this case, the master (microcontroller A) generates the timing and terminates the transfer.

The possibility of connecting more than one microcontroller to the I<sup>2</sup>C-bus means that

## The I<sup>2</sup>C-bus and how to use it

more than one master could try to initiate a data transfer at the same time. To avoid the chaos that might ensue from such an event – an arbitration procedure has been developed. This procedure relies on the wired-AND connection of all I<sup>2</sup>C interfaces to the I<sup>2</sup>C-bus.

If two or more masters try to put information onto the bus, the first to produce a 'one' while the other produces a 'zero' will lose the arbitration. The clock signals during arbitration are a synchronized combination of the clocks generated by the masters using the wired-AND connection to the SCL line (for more detailed information concerning arbitration see Section 7.0).

Generation of clock signals on the I<sup>2</sup>C-bus is always the responsibility of master devices; each master generates its own clock signals when transferring data on the bus. Bus clock signals from a master can only be altered when they are stretched by a slow-slave device holding-down the clock line, or by another master when arbitration occurs.

### 4.0 GENERAL CHARACTERISTICS

Both SDA and SCL are bidirectional lines, connected to a positive supply voltage via a pull-up resistor (see Fig.4). When the bus is free, both lines are HIGH. The output stages of devices connected to the bus must have an open-drain or open-collector in order to perform the wired-AND function. Data on the I<sup>2</sup>C-bus can be transferred at a rate up to 100 kbit/s in the standard-mode, or up to 400 kbit/s in the fast-mode. The number of interfaces connected to the bus is solely dependent on the bus capacitance limit of 400 pF.

### 5.0 BIT TRANSFER

Due to the variety of different technology devices (CMOS, NMOS, bipolar) which can be connected to the I<sup>2</sup>C-bus, the levels of the logical '0' (LOW) and '1' (HIGH) are not fixed and depend on the associated level of V<sub>DD</sub> (see Section 15.0 for Electrical Specifications). One clock pulse is generated for each data bit transferred.

### 5.1 Data Validity

The data on the SDA line must be stable

during the HIGH period of the clock. The HIGH or LOW state of the data line can only change when the clock signal on the SCL line is LOW (see Fig.5).

### 5.2 START and STOP Conditions

Within the procedure of the I<sup>2</sup>C-bus, unique situations arise which are defined as START and STOP conditions (see Fig.6).

A HIGH to LOW transition on the SDA line while SCL is HIGH is one such unique case. This situation indicates a START condition.

A LOW to HIGH transition on the SDA line while SCL is HIGH defines a STOP condition.

START and STOP conditions are always generated by the master. The bus is considered to be busy after the START condition. The bus is considered to be free again a certain time after the STOP condition. This bus free situation is specified in Section 15.0.

Detection of START and STOP conditions by devices connected to the bus is easy if they incorporate the necessary interfacing hardware. However, microcontrollers with no such interface have to sample the SDA line at least twice per clock period in order to sense the transition.

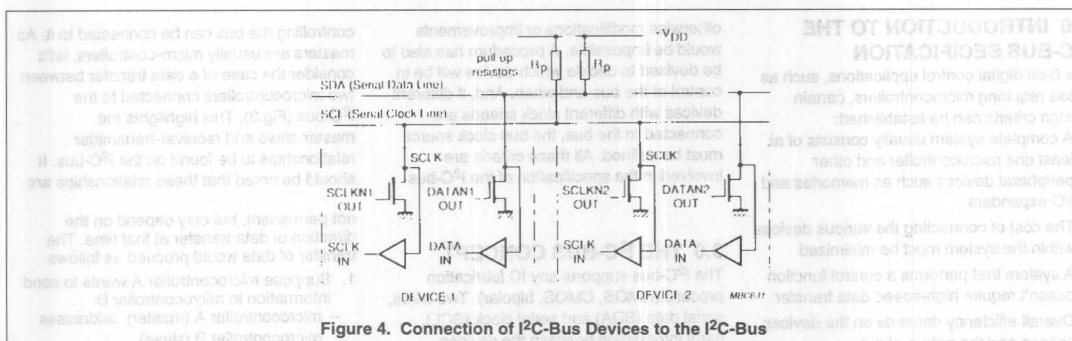


Figure 4. Connection of I<sup>2</sup>C-Bus Devices to the I<sup>2</sup>C-Bus

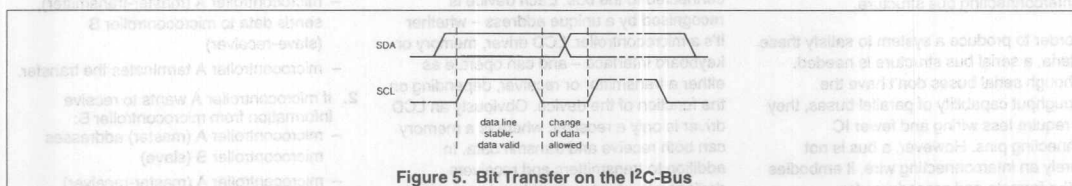


Figure 5. Bit Transfer on the I<sup>2</sup>C-Bus

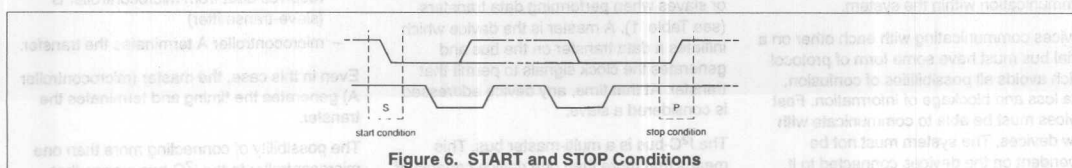


Figure 6. START and STOP Conditions



## The I<sup>2</sup>C-bus and how to use it

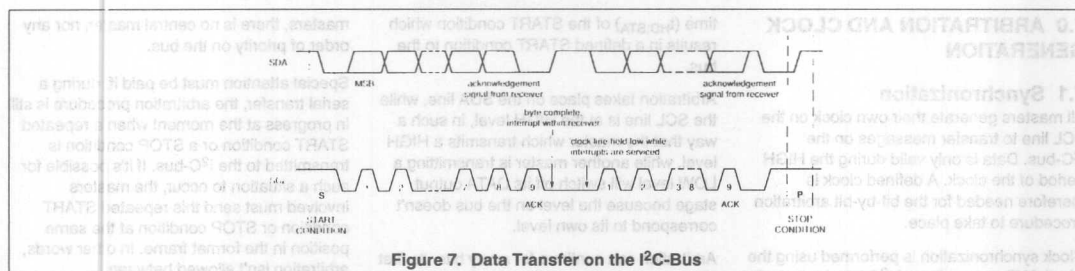
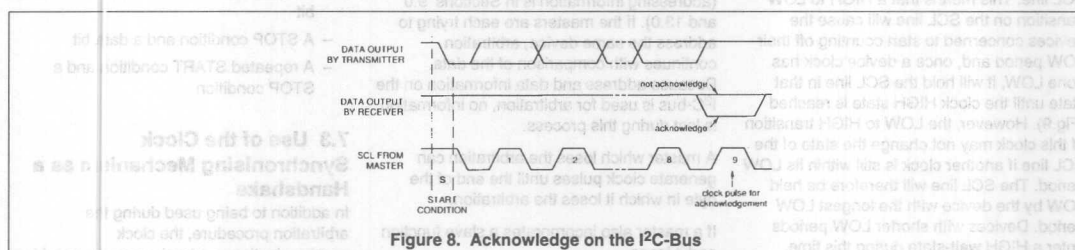
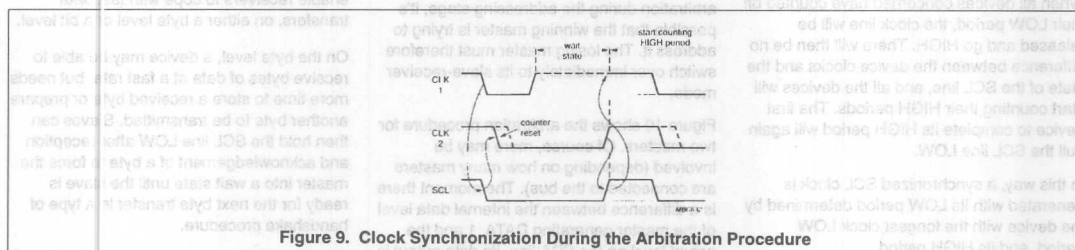
Figure 7. Data Transfer on the I<sup>2</sup>C-BusFigure 8. Acknowledge on the I<sup>2</sup>C-Bus

Figure 9. Clock Synchronization During the Arbitration Procedure

## 6.0 TRANSFERRING DATA

### 6.1 Byte Format

Every byte put on the SDA line must be 8-bits long. The number of bytes that can be transmitted per transfer is unrestricted. Each byte has to be followed by an acknowledge bit. Data is transferred with the most significant bit (MSB) first (Fig.7). If a receiver can't receive another complete byte of data until it has performed some other function, for example servicing an internal interrupt, it can hold the clock line SCL LOW to force the transmitter into a wait state. Data transfer then continues when the receiver is ready for another byte of data and releases clock line SCL.

In some cases, it's permitted to use a different format from the I<sup>2</sup>C-bus format (for CBUS compatible devices for example). A message which starts with such an address can be terminated by generation of a STOP

condition, even during the transmission of a byte. In this case, no acknowledge is generated (see Section 9.1.3).

### 6.2 Acknowledge

Data transfer with acknowledge is obligatory. The acknowledge-related clock pulse is generated by the master. The transmitter releases the SDA line (HIGH) during the acknowledge clock pulse.

The receiver must pull down the SDA line during the acknowledge clock pulse so that it remains stable LOW during the HIGH period of this clock pulse (Fig.8). Of course, set-up and hold times (specified in Section 15) must also be taken into account.

Usually, a receiver which has been addressed is obliged to generate an acknowledge after each byte has been received, except when the message starts with a CBUS address (see Section 9.1.3).

When a slave-receiver doesn't acknowledge the slave address (for example, it's unable to receive because it's performing some real-time function), the data line must be left HIGH by the slave. The master can then generate a STOP condition to abort the transfer.

If a slave-receiver does acknowledge the slave address but, some time later in the transfer cannot receive any more data bytes, the master must again abort the transfer. This is indicated by the slave generating the not acknowledge on the first byte to follow. The slave leaves the data line HIGH and the master generates the STOP condition.

If a master-receiver is involved in a transfer, it must signal the end of data to the slave-transmitter by not generating an acknowledge on the last byte that was clocked out of the slave. The slave-transmitter must release the data line to allow the master to generate the STOP condition.

## The I<sup>2</sup>C-bus and how to use it

### 7.0 ARBITRATION AND CLOCK GENERATION

#### 7.1 Synchronization

All masters generate their own clock on the SCL line to transfer messages on the I<sup>2</sup>C-bus. Data is only valid during the HIGH period of the clock. A defined clock is therefore needed for the bit-by-bit arbitration procedure to take place.

Clock synchronization is performed using the wired-AND connection of I<sup>2</sup>C interfaces to the SCL line. This means that a HIGH to LOW transition on the SCL line will cause the devices concerned to start counting off their LOW period and, once a device clock has gone LOW, it will hold the SCL line in that state until the clock HIGH state is reached (Fig.9). However, the LOW to HIGH transition of this clock may not change the state of the SCL line if another clock is still within its LOW period. The SCL line will therefore be held LOW by the device with the longest LOW period. Devices with shorter LOW periods enter a HIGH wait-state during this time.

When all devices concerned have counted off their LOW period, the clock line will be released and go HIGH. There will then be no difference between the device clocks and the state of the SCL line, and all the devices will start counting their HIGH periods. The first device to complete its HIGH period will again pull the SCL line LOW.

In this way, a synchronized SCL clock is generated with its LOW period determined by the device with the longest clock LOW period, and its HIGH period determined by the one with the shortest clock HIGH period.

#### 7.2 Arbitration

A master may start a transfer only if the bus is free. Two or more masters may generate a START condition within the minimum hold

time ( $t_{HD,STA}$ ) of the START condition which results in a defined START condition to the bus.

Arbitration takes place on the SDA line, while the SCL line is at the HIGH level, in such a way that the master which transmits a HIGH level, while another master is transmitting a LOW level will switch off its DATA output stage because the level on the bus doesn't correspond to its own level.

Arbitration can continue for many bits. Its first stage is comparison of the address bits (addressing information is in Sections 9.0 and 13.0). If the masters are each trying to address the same device, arbitration continues with comparison of the data. Because address and data information on the I<sup>2</sup>C-bus is used for arbitration, no information is lost during this process.

A master which loses the arbitration can generate clock pulses until the end of the byte in which it loses the arbitration.

If a master also incorporates a slave function and it loses

arbitration during the addressing stage, it's possible that the winning master is trying to address it. The losing master must therefore switch over immediately to its slave-receiver mode.

Figure 10 shows the arbitration procedure for two masters. Of course, more may be involved (depending on how many masters are connected to the bus). The moment there is a difference between the internal data level of the master generating DATA 1 and the actual level on the SDA line, its data output is switched off, which means that a HIGH output level is then connected to the bus. This will not affect the data transfer initiated by the winning master.

Since control of the I<sup>2</sup>C-bus is decided solely on the address and data sent by competing

masters, there is no central master, nor any order of priority on the bus.

Special attention must be paid if, during a serial transfer, the arbitration procedure is still in progress at the moment when a repeated START condition or a STOP condition is transmitted to the I<sup>2</sup>C-bus. If it's possible for such a situation to occur, the masters involved must send this repeated START condition or STOP condition at the same position in the format frame. In other words, arbitration isn't allowed between:

- A repeated START condition and a data bit
- A STOP condition and a data bit
- A repeated START condition and a STOP condition.

#### 7.3 Use of the Clock Synchronising Mechanism as a Handshake

In addition to being used during the arbitration procedure, the clock synchronization mechanism can be used to enable receivers to cope with fast data transfers, on either a byte level or a bit level.

On the byte level, a device may be able to receive bytes of data at a fast rate, but needs more time to store a received byte or prepare another byte to be transmitted. Slaves can then hold the SCL line LOW after reception and acknowledgement of a byte to force the master into a wait state until the slave is ready for the next byte transfer in a type of handshake procedure.

On the bit level, a device such as a microcontroller without, or with only a limited hardware I<sup>2</sup>C interface on-chip can slow down the bus clock by extending each clock LOW period. The speed of any master is thereby adapted to the internal operating rate of this device.

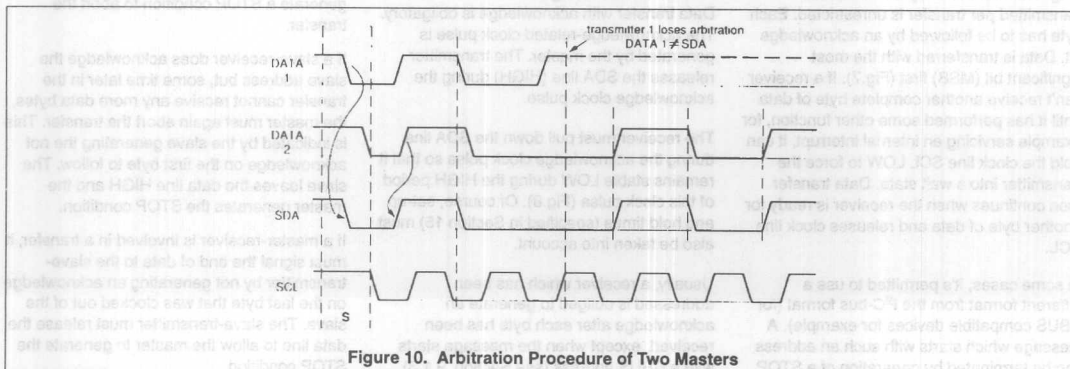


Figure 10. Arbitration Procedure of Two Masters

The I<sup>2</sup>C-bus and how to use it

## 8.0 FORMATS WITH 7-BIT ADDRESSES

Data transfers follow the format shown in Fig.11. After the START condition (S), a slave address is sent. This address is 7 bits long followed by an eighth bit which is a data direction bit (R/W) – a 'zero' indicates a transmission (WRITE), a 'one' indicates a request for data (READ). A data transfer is always terminated by a STOP condition (P) generated by the master. However, if a master still wishes to communicate on the bus, it can generate a repeated START condition (S) and address another slave without first generating a STOP condition. Various combinations of read/write formats are then possible within such a transfer.

Possible data transfer formats are:

- Master-transmitter transmits to slave-receiver. The transfer direction is not changed (Fig.12)
- Master reads slave immediately after first byte (Fig.13). At the moment of the first acknowledge, the master-transmitter becomes a master-receiver and the slave-receiver becomes a slave-transmitter. This acknowledge is still generated by the slave. The STOP condition is generated by the master
- Combined format (Fig.14). During a change of direction within a transfer, the START condition and the slave address are both repeated, but with the R/W bit reversed.

## NOTES:

1. Combined formats can be used, for example, to control a serial memory. During the first data byte, the internal memory location has to be written. After the START condition and slave address is repeated, data can be transferred.
2. All decisions on auto-increment or decrement of previously accessed memory locations etc. are taken by the designer of the device.
3. Each byte is followed by an acknowledgement bit as indicated by the A or  $\bar{A}$  blocks in the sequence.
4. I<sup>2</sup>C-bus compatible devices must reset their bus logic on receipt of a START or repeated START condition such that they all anticipate the sending of a slave address.

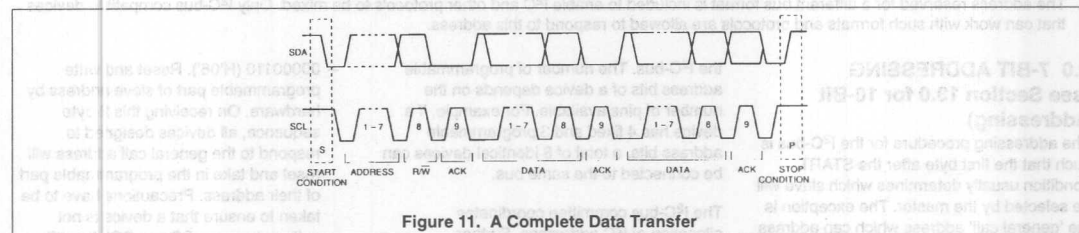


Figure 11. A Complete Data Transfer

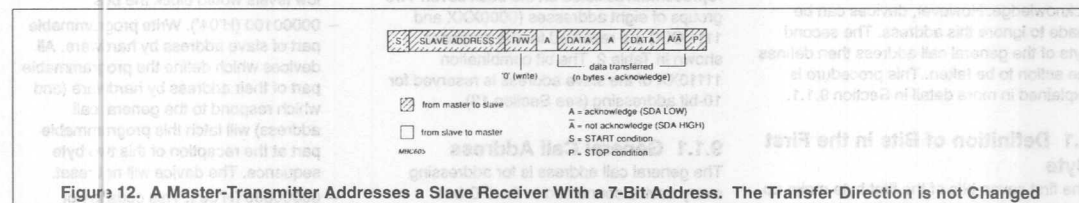


Figure 12. A Master-Transmitter Addresses a Slave Receiver With a 7-Bit Address. The Transfer Direction is not Changed

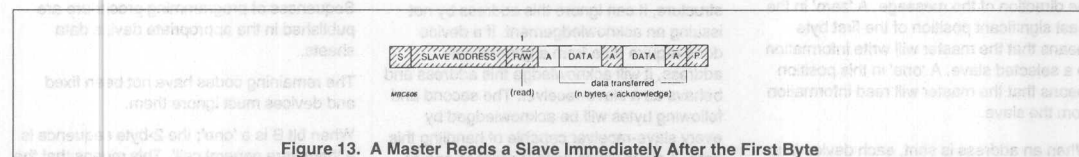


Figure 13. A Master Reads a Slave Immediately After the First Byte

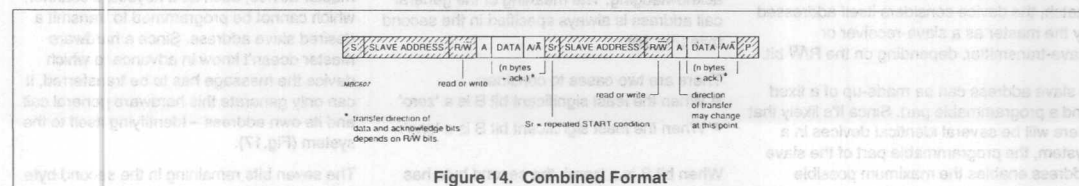


Figure 14. Combined Format

# The I<sup>2</sup>C-bus and how to use it

Table 2. Definition of Bits in the First Byte

Slave address	R/ bit	Description
0000 000	0	General call address
0000 000	1	START byte
0000 001	X	CBUS address
0000 010	X	Address reserved for different bus format
0000 011	X	Reserved for future purposes
0000 1XX	X	Reserved for future purposes
1111 1XX	X	Reserved for future purposes
1111 0XX	X	10-bit slave addressing

NOTES:

1. No device is allowed to acknowledge at the reception of the START byte.
2. The CBUS address has been reserved to enable the inter-mixing of CBUS compatible and I<sup>2</sup>C-bus compatible devices in the same system. I<sup>2</sup>C-bus compatible devices are not allowed to respond on reception of this address.
3. The address reserved for a different bus format is included to enable I<sup>2</sup>C and other protocols to be mixed. Only I<sup>2</sup>C-bus compatible devices that can work with such formats and protocols are allowed to respond to this address.

## 9.0 7-BIT ADDRESSING (see Section 13.0 for 10-Bit Addressing)

The addressing procedure for the I<sup>2</sup>C-bus is such that the first byte after the START condition usually determines which slave will be selected by the master. The exception is the 'general call' address which can address all devices. When this address is used, all devices should, in theory, respond with an acknowledge. However, devices can be made to ignore this address. The second byte of the general call address then defines the action to be taken. This procedure is explained in more detail in Section 9.1.1.

### 9.1 Definition of Bits in the First Byte

The first seven bits of the first byte make up the slave address (Fig.15). The eighth bit is the LSB (least significant bit). It determines the direction of the message. A 'zero' in the least significant position of the first byte means that the master will write information to a selected slave. A 'one' in this position means that the master will read information from the slave.

When an address is sent, each device in a system compares the first seven bits after the START condition with its address. If they match, the device considers itself addressed by the master as a slave-receiver or slave-transmitter, depending on the R/W bit.

A slave address can be made-up of a fixed and a programmable part. Since it's likely that there will be several identical devices in a system, the programmable part of the slave address enables the maximum possible number of such devices to be connected to

the I<sup>2</sup>C-bus. The number of programmable address bits of a device depends on the number of pins available. For example, if a device has 4 fixed and 3 programmable address bits, a total of 8 identical devices can be connected to the same bus.

The I<sup>2</sup>C-bus committee coordinates allocation of I<sup>2</sup>C addresses. Further information can be obtained from the Philips representatives listed on the back cover. Two groups of eight addresses (0000XXX and 1111XXX) are reserved for the purposes shown in Table 2. The bit combination 11110XX of the slave address is reserved for 10-bit addressing (see Section 13).

#### 9.1.1 General Call Address

The general call address is for addressing every device connected to the I<sup>2</sup>C-bus. However, if a device doesn't need any of the data supplied within the general call structure, it can ignore this address by not issuing an acknowledgement. If a device does require data from a general call address, it will acknowledge this address and behave as a slave-receiver. The second and following bytes will be acknowledged by every slave-receiver capable of handling this data. A slave which cannot process one of these bytes must ignore it by not acknowledging. The meaning of the general call address is always specified in the second byte (Fig.16).

There are two cases to consider:

- When the least significant bit B is a 'zero'
- When the least significant bit B is a 'one'.

When bit B is a 'zero', the second byte has the following definition:

- 00000110 (H'06'). Reset and write programmable part of slave address by hardware. On receiving this 2-byte sequence, all devices designed to respond to the general call address will reset and take in the programmable part of their address. Precautions have to be taken to ensure that a device is not pulling down the SDA or SCL line after applying the supply voltage, since these low levels would block the bus
- 00000100 (H'04'). Write programmable part of slave address by hardware. All devices which define the programmable part of their address by hardware (and which respond to the general call address) will latch this programmable part at the reception of this two byte sequence. The device will not reset.
- 00000000 (H'00'). This code is not allowed to be used as the second byte.

Sequences of programming procedure are published in the appropriate device data sheets.

The remaining codes have not been fixed and devices must ignore them.

When bit B is a 'one', the 2-byte sequence is a 'hardware general call'. This means that the sequence is transmitted by a hardware master device, such as a keyboard scanner, which cannot be programmed to transmit a desired slave address. Since a hardware master doesn't know in advance to which device the message has to be transferred, it can only generate this hardware general call and its own address - identifying itself to the system (Fig.17).

The seven bits remaining in the second byte contain the address of the hardware master.



## The I<sup>2</sup>C-bus and how to use it

This address is recognised by an intelligent device (e.g. a microcontroller) connected to the bus which will then direct the information from the hardware master. If the hardware master can also act as a slave, the slave address is identical to the master address.

In some systems, an alternative could be that the hardware master transmitter is set in the slave-receiver mode after the system reset. In this way, a system configuring master can tell the hardware master-transmitter (which is

now in slave-receiver mode) to which address data must be sent (Fig.18). After this programming procedure, the hardware master remains in the master-transmitter mode.

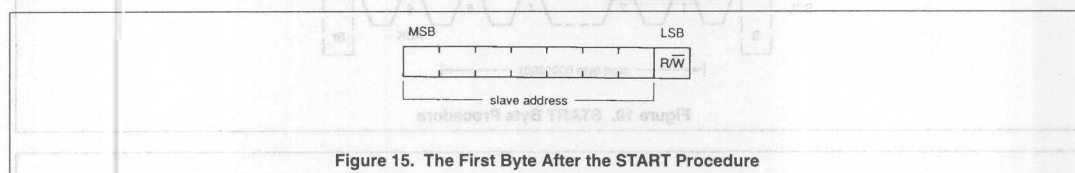


Figure 15. The First Byte After the START Procedure

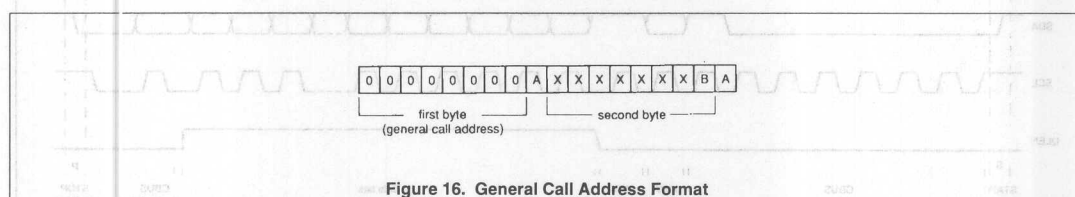


Figure 16. General Call Address Format

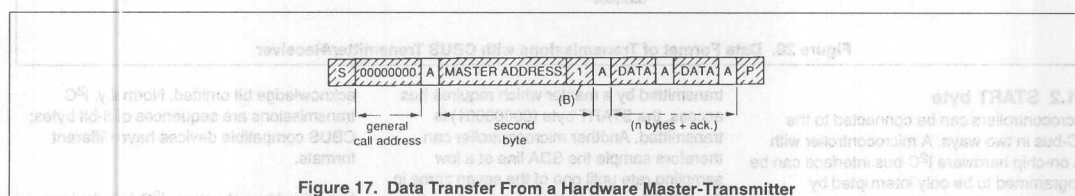
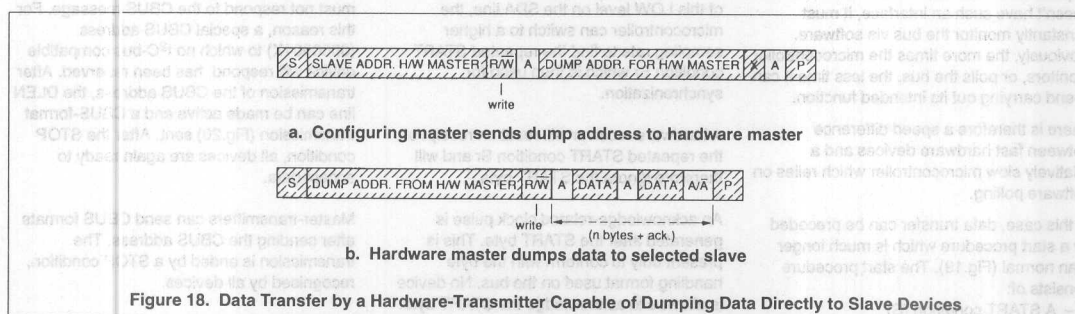


Figure 17. Data Transfer From a Hardware Master-Transmitter

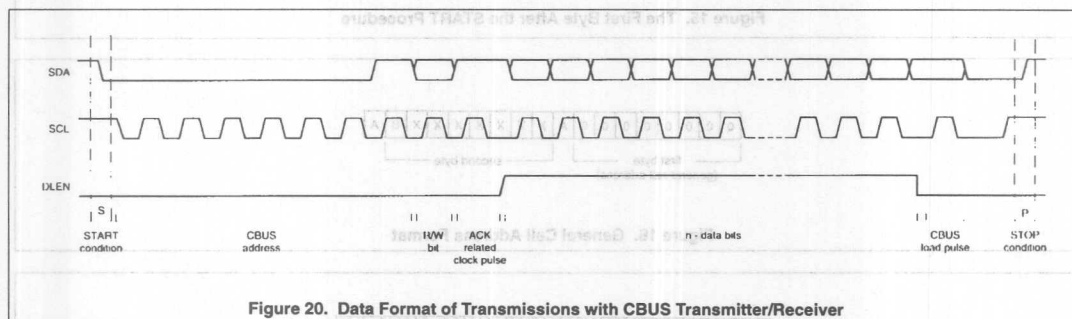
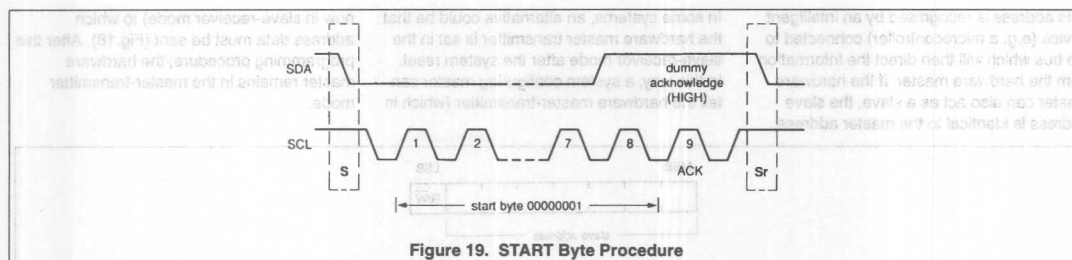


**Figure 18. Data Transfer by a Hardware-Transmitter Capable of Dumping Data Directly to Slave Devices**



## The I<sup>2</sup>C-bus and how to use it

## The I<sup>2</sup>C-bus and how to use it



### 9.1.2 START byte

Microcontrollers can be connected to the I<sup>2</sup>C-bus in two ways. A microcontroller with an on-chip hardware I<sup>2</sup>C-bus interface can be programmed to be only interrupted by requests from the bus. When the device doesn't have such an interface, it must constantly monitor the bus via software. Obviously, the more times the microcontroller monitors, or polls the bus, the less time it can spend carrying out its intended function.

There is therefore a speed difference between fast hardware devices and a relatively slow microcontroller which relies on software polling.

In this case, data transfer can be preceded by a start procedure which is much longer than normal (Fig.19). The start procedure consists of:

- A START condition (S)
- A START byte (00000001)
- An acknowledge clock pulse (ACK)
- A repeated START condition (Sr).

After the START condition S has been

transmitted by a master which requires bus access, the START byte (00000001) is transmitted. Another microcontroller can therefore sample the SDA line at a low sampling rate until one of the seven zeros in the START byte is detected. After detection of this LOW level on the SDA line, the microcontroller can switch to a higher sampling rate to find the repeated START condition Sr which is then used for synchronization.

A hardware receiver will reset on receipt of the repeated START condition Sr and will therefore ignore the START byte.

An acknowledge-related clock pulse is generated after the START byte. This is present only to conform with the byte handling format used on the bus. No device is allowed to acknowledge the START byte.

### 9.1.3 CBUS Compatibility

CBUS receivers can be connected to the I<sup>2</sup>C-bus. However, a third bus line called DLEN must then be connected and the

acknowledge bit omitted. Normally, I<sup>2</sup>C transmissions are sequences of 8-bit bytes; CBUS compatible devices have different formats.

In a mixed bus structure, I<sup>2</sup>C-bus devices must not respond to the CBUS message. For this reason, a special CBUS address (0000001X) to which no I<sup>2</sup>C-bus compatible device will respond, has been reserved. After transmission of the CBUS address, the DLEN line can be made active and a CBUS-format transmission (Fig.20) sent. After the STOP condition, all devices are again ready to accept data.

Master-transmitters can send CBUS formats after sending the CBUS address. The transmission is ended by a STOP condition, recognised by all devices.

**NOTE:** If the CBUS configuration is known, and expansion with CBUS compatible devices isn't foreseen, the designer is allowed to adapt the hold time to the specific requirements of the device(s) used.

# The I<sup>2</sup>C-bus and how to use it

## 10.0 ELECTRICAL CHARACTERISTICS FOR I<sup>2</sup>C-BUS DEVICES

The electrical specifications for the I/Os of I<sup>2</sup>C-bus devices and the characteristics of the bus lines connected to them are given in Tables 3 and 4 in Section-15.

I<sup>2</sup>C-bus devices with fixed input levels of 1.5 V and 3 V can each have their own appropriate supply voltage. Pull-up resistors

must be connected to a 5 V  $\pm$  10% supply (Fig.21). I<sup>2</sup>C-bus devices with input levels related to V<sub>DD</sub> must have one common supply line to which the pull-up resistor is also connected (Fig.22).

When devices with fixed input levels are mixed with devices with input levels related to V<sub>DD</sub>, the latter devices must be connected to one common supply line of 5 V  $\pm$  10% and must have pull-up resistors connected to their SDA and SCL pins as shown in Fig.23.

Input levels are defined in such a way that:

- The noise margin on the LOW level is 0.1 V<sub>DD</sub>
- The noise margin on the HIGH level is 0.2 V<sub>DD</sub>
- As shown in Fig.24, series resistors (R<sub>S</sub>) of e.g. 300  $\Omega$  can be used for protection against high-voltage spikes on the SDA and SCL lines (due to flash-over of a TV picture tube, for example).

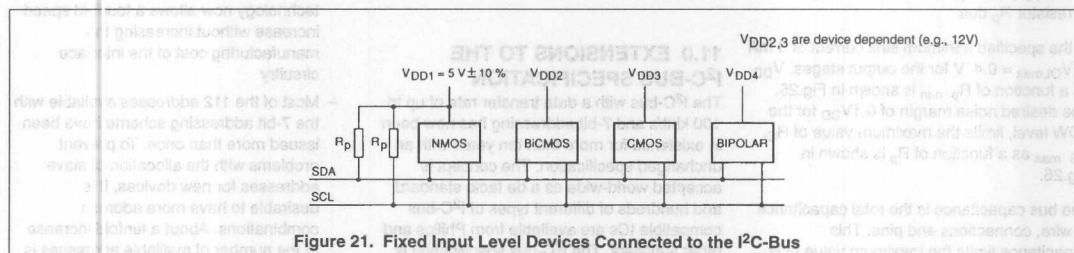


Figure 21. Fixed Input Level Devices Connected to the I<sup>2</sup>C-Bus

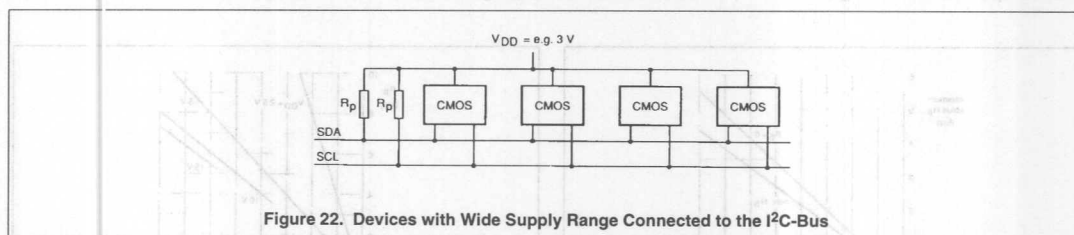


Figure 22. Devices with Wide Supply Range Connected to the I<sup>2</sup>C-Bus

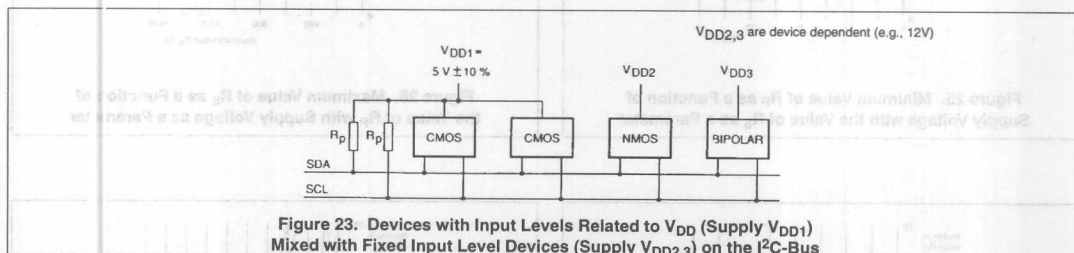


Figure 23. Devices with Input Levels Related to V<sub>DD</sub> (Supply V<sub>DD1</sub>) Mixed with Fixed Input Level Devices (Supply V<sub>DD2,3</sub>) on the I<sup>2</sup>C-Bus

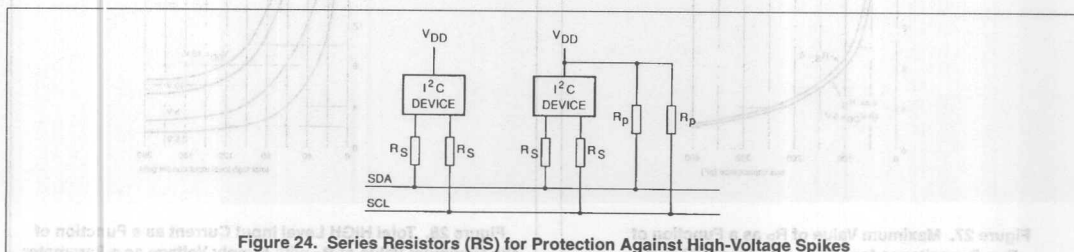


Figure 24. Series Resistors (R<sub>S</sub>) for Protection Against High-Voltage Spikes

## The I<sup>2</sup>C-bus and how to use it

### 10.1 Maximum and minimum values of resistors $R_P$ and $R_S$

For standard-mode I<sup>2</sup>C-bus devices, the values of resistors  $R_P$  and  $R_S$  in Fig.24 depend on the following parameters:

- Supply voltage
- Bus capacitance
- Number of connected devices (input current + leakage current).

The supply voltage limits the minimum value of resistor  $R_P$  due

to the specified minimum sink current of 3 mA at  $V_{OLmax} = 0.4$  V for the output stages.  $V_{DD}$  as a function of  $R_P$  min is shown in Fig.25. The desired noise margin of  $0.1V_{DD}$  for the LOW level, limits the maximum value of  $R_S$ .  $R_S$  max as a function of  $R_P$  is shown in Fig.26.

The bus capacitance is the total capacitance of wire, connections and pins. This capacitance limits the maximum value of  $R_P$  due to the specified rise time. Fig.27 shows

$R_P$  max as a function of bus capacitance.

The maximum HIGH level input current of each input/output connection has a specified maximum value of  $10 \mu A$ . Due to the desired noise margin of  $0.2V_{DD}$  for the HIGH level, this input current limits the maximum value of  $R_P$ . This limit depends on  $V_{DD}$ . The total HIGH level input current is shown as a function of  $R_P$  max in Fig.28.

### 11.0 EXTENSIONS TO THE I<sup>2</sup>C-BUS SPECIFICATION

The I<sup>2</sup>C-bus with a data transfer rate of up to 100 kbit/s and 7-bit addressing has now been in existence for more than ten years with an unchanged specification. The concept is accepted world-wide as a de facto standard and hundreds of different types of I<sup>2</sup>C-bus compatible ICs are available from Philips and other suppliers. The I<sup>2</sup>C-bus specification is now extended with the following two features:

- A fast-mode which allows a fourfold increase of the bit rate to 0 to 400 kbit/s
- 10-bit addressing which allows the use of up to 1024 additional addresses.

There are two reasons for these extensions to the I<sup>2</sup>C-bus specification:

- New applications will need to transfer a larger amount of serial data and will therefore demand a higher bit rate than 100 kbit/s. Improved IC manufacturing technology now allows a fourfold speed increase without increasing the manufacturing cost of the interface circuitry
- Most of the 112 addresses available with the 7-bit addressing scheme have been issued more than once. To prevent problems with the allocation of slave addresses for new devices, it is desirable to have more address combinations. About a tenfold increase of the number of available addresses is obtained with the new 10-bit addressing.

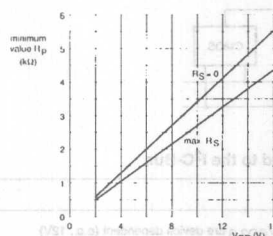


Figure 25. Minimum Value of  $R_P$  as a Function of Supply Voltage with the Value of  $R_S$  as a Parameter

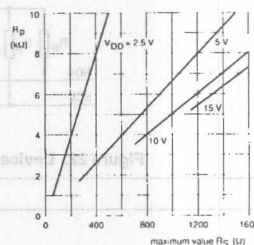


Figure 26. Maximum Value of  $R_S$  as a Function of the Value of  $R_P$  with Supply Voltage as a Parameter

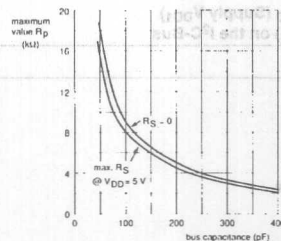


Figure 27. Maximum Value of  $R_P$  as a Function of Bus Capacitance for a Standard-Mode I<sup>2</sup>C-Bus

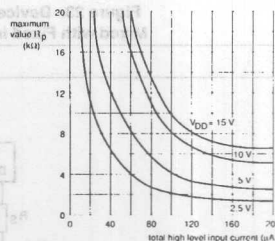


Figure 28. Total HIGH Level Input Current as a Function of the Maximum Value of  $R_P$  with Supply Voltage as a Parameter

## The I<sup>2</sup>C-bus and how to use it

All new devices with an I<sup>2</sup>C-bus interface are provided with the fast-mode. Preferably, they should be able to receive and/or transmit at 400 kbit/s. The minimum requirement is that they can synchronize with a 400 kbit/s transfer; they can then prolong the LOW period of the SCL signal to slow down the transfer. Fast-mode devices must be downward-compatible which means that they must still be able to communicate with 0 to 100 kbit/s devices in a 0 to 100 kbit/s I<sup>2</sup>C-bus system.

Obviously, devices with a 0 to 100 kbit/s I<sup>2</sup>C-bus interface cannot be incorporated in a fast-mode I<sup>2</sup>C-bus system because, since they cannot follow the higher transfer rate, unpredictable states of these devices would occur.

Slave devices with a fast-mode I<sup>2</sup>C-bus interface can have a 7-bit or a 10-bit slave address. However, a 7-bit address is preferred because it is the cheapest solution in hardware and it results in the shortest message length. Devices with 7-bit and 10-bit addresses can be mixed in the same I<sup>2</sup>C-bus system regardless of whether it is a 0 to 100 kbit/s standard-mode system or a 0 to 400 kbit/s fast-mode system. Both existing and future masters can generate either 7-bit or 10-bit addresses.

### 12.0 FAST-MODE

In the fast-mode of the I<sup>2</sup>C-bus, the protocol, format, logic levels and maximum capacitive load for the SDA and SCL lines quoted in the previous I<sup>2</sup>C-bus specification are unchanged. Changes to the previous I<sup>2</sup>C-bus specification are:

- The maximum bit rate is increased to 400 kbit/s
- Timing of the serial data (SDA) and serial clock (SCL) signals has been adapted. There is no need for compatibility with other bus systems such as CBUS because they cannot operate at the increased bit rate
- The inputs of fast-mode devices must incorporate spike suppression and a Schmitt trigger at the SDA and SCL inputs
- The output buffers of fast-mode devices must incorporate slope control of the falling edges of the SDA and SCL signals
- If the power supply to a fast-mode device is switched off, the SDA and SCL I/O pins must be floating so that they don't obstruct the bus lines
- The external pull-up devices connected to the bus lines must be adapted to accommodate the shorter maximum

permissible rise time for the fast-mode I<sup>2</sup>C-bus. For bus loads up to 200 pF, the pull-up device for each bus line can be a resistor; for bus loads between 200 pF and 400 pF, the pull-up device can be a current source (3mA max.) or a switched resistor circuit as shown in Fig.37.

### 13.0 10-BIT ADDRESSING

The 10-bit addressing does not change the format in the I<sup>2</sup>C-bus specification. Using 10 bits for addressing exploits the reserved combination 1111XXX for the first seven bits of the first byte following a START (S) or repeated START (Sr) condition as explained in Section 9.1. The 10-bit addressing does not affect the existing 7-bit addressing.

Devices with 7-bit and 10-bit addresses can be connected to the same I<sup>2</sup>C-bus, and both 7-bit and 10-bit addressing can be used in a standard-mode system (up to 100 kbit/s) or a fast-mode system (up to 400 kbit/s).

Although there are eight possible combinations of the reserved address bits 1111XXX, only the four combinations 11110XX are used for 10-bit addressing. The remaining four combinations 11111XX are reserved for future I<sup>2</sup>C-bus enhancements.

#### 13.1 Definition of Bits in the First Two Bytes

The 10-bit slave address is formed from the first two bytes following a START condition (S) or a repeated START condition (Sr).

The first seven bits of the first byte are the combination 11110XX of which the last two bits (XX) are the two most-significant bits (MSBs) of the 10-bit address; the eighth bit of the first byte is the R/W bit that determines the direction of the message. A 'zero' in the least significant position of the first byte means that the master will write information to a selected slave. A 'one' in this position means that the master will read information from the slave.

If the R/W bit is 'zero', then the second byte contains the remaining 8 bits (XXXXXXXX) of the 10-bit address. If the R/W bit is 'one', then the next byte contains data transmitted from a slave to a master.

#### 13.2 Formats with 10-bit Addresses

Various combinations of read/write formats are possible within a transfer that includes 10-bit addressing. Possible data transfer formats are:

- **Master-transmitter transmits to slave-receiver with a 10-bit slave address. The transfer direction is not changed (Fig.29).** When a 10-bit

address follows a START condition, each slave compares the first seven bits of the first byte of the slave address (11110XX) with its own address and tests if the eighth bit (R/W direction bit) is 0. It is possible that more than one device will find a match and generate an acknowledge (A1). All slaves that found a match will compare the eight bits of the second byte of the slave address (XXXXXXXX) with their own addresses, but only one slave will find a match and generate an acknowledge (A2). The matching slave will remain addressed by the master until it receives a STOP condition (P) or a repeated START condition (Sr) followed by a different slave address

#### NOTES:

1. Combined formats can be used, for example, to control a serial memory. During the first data byte, the internal memory location has to be written. After the START condition and slave address is repeated, data can be transferred.
2. All decisions on auto-increment or decrement of previously accessed memory locations etc. are taken by the designer of the device.
3. Each byte is followed by an acknowledgement bit as indicated by the A or  $\bar{A}$  blocks in the sequence.
4. I<sup>2</sup>C-bus compatible devices must reset their bus logic on receipt of a START or repeated START condition such that they all anticipate the sending of a slave address.

- **Master-receiver reads slave-transmitter with a 10-bit slave address. The transfer direction is changed after the second R/W bit (Fig.30).** Up to and including acknowledge bit A2, the procedure is the same as that described for a master-transmitter addressing a slave-receiver. After the repeated START condition (Sr), a matching slave remembers that it was addressed before. This slave then checks if the first seven bits of the first byte of the slave address following Sr are the same as they were after the START condition (S), and tests if the eighth (R/W) bit is 1. If there is a match, the slave considers that it has been addressed as a transmitter and generates acknowledge A3. The slave-transmitter remains addressed until it receives a STOP condition (P) or until it receives another repeated START condition (Sr) followed by a different slave address. After a repeated START condition (Sr), all the other slave devices will also compare the first seven bits of the first byte of the slave address (11110XX) with their own addresses and



## The I<sup>2</sup>C-bus and how to use it

test the eighth (R/W) bit. However, none of them will be addressed because R/W = 1 (for 10-bit devices), or the 11110XX slave address (for 7-bit devices) does not match)

- **Combined format. A master transmits data to a slave and then reads data from the same slave (Fig.31).** The same master occupies the bus all the

time. The transfer direction is changed after the second R/W bit

- **Combined format. A master transmits data to one slave and then transmits data to another slave (Fig.32).** The same master occupies the bus all the time
- **Combined format. 10-bit and 7-bit addressing combined in one serial**

**transfer (Fig.33).** After each START condition (S), or each repeated START condition (Sr), a 10-bit or 7-bit slave address can be transmitted. Figure 33 shows how a master-transmits data to a slave with a 7-bit address and then transmits data to a second slave with a 10-bit address. The same master occupies the bus all the time.

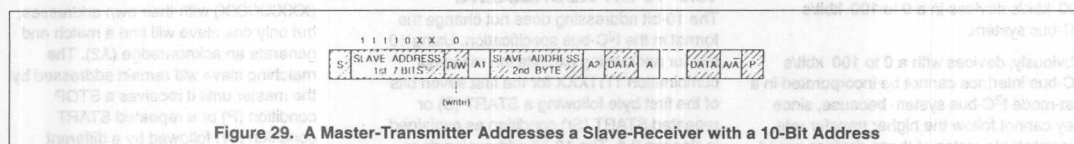


Figure 29. A Master-Transmitter Addresses a Slave-Receiver with a 10-Bit Address

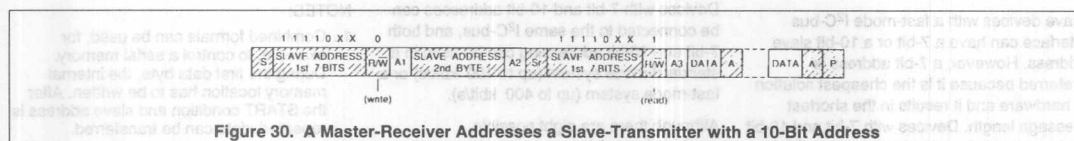


Figure 30. A Master-Receiver Addresses a Slave-Transmitter with a 10-Bit Address

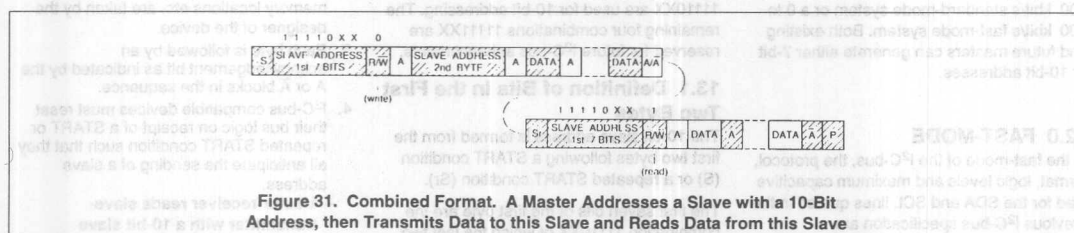


Figure 31. Combined Format. A Master Addresses a Slave with a 10-Bit Address, then Transmits Data to this Slave and Reads Data from this Slave

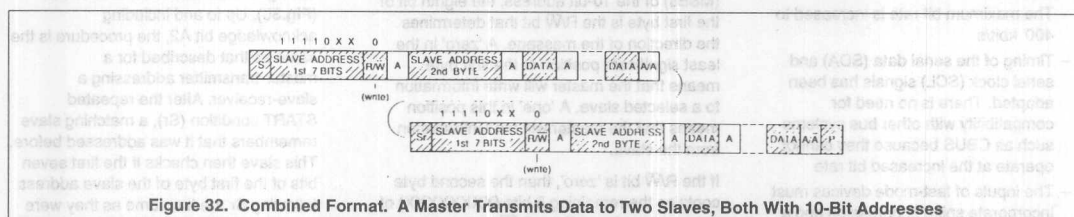


Figure 32. Combined Format. A Master Transmits Data to Two Slaves, Both With 10-Bit Addresses

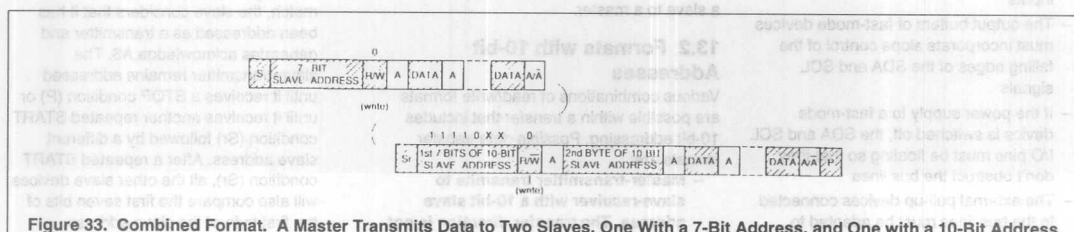
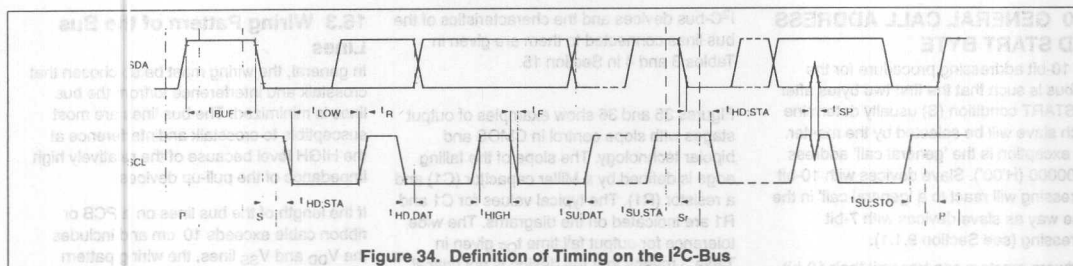


Figure 33. Combined Format. A Master Transmits Data to Two Slaves, One With a 7-Bit Address, and One with a 10-Bit Address.



The I<sup>2</sup>C-bus and how to use itFigure 34. Definition of Timing on the I<sup>2</sup>C-BusTable 3. Characteristics of the SDA and SCL I/O Stages for I<sup>2</sup>C-Bus Devices

Parameter	Symbol	standard-mode devices		fast-mode devices		Unit
		Min.	Max.	Min.	Max.	
LOW level input voltage: fixed input levels V <sub>DD</sub> -related input levels	V <sub>IL</sub>	-0.5 -0.5	1.5 0.3V <sub>DD</sub>	-0.5 -0.5	1.5 0.3V <sub>DD</sub>	V
HIGH level input voltage: fixed input levels V <sub>DD</sub> -related input levels	V <sub>IH</sub>	3.0 0.7V <sub>DD</sub>	*1) *1)	3.0 0.7V <sub>DD</sub>	*1) *1)	V
Hysteresis of Schmitt trigger inputs: fixed input levels V <sub>DD</sub> -related input levels	V <sub>hys</sub>	n/a n/a	n/a n/a	0.2 0.05V <sub>DD</sub>	- -	V
Pulse width of spikes which must be suppressed by the input filter	t <sub>SP</sub>	n/a	n/a	0	50	ns
LOW level output voltage (open drain or open collector): at 3 mA sink current at 6 mA sink current	V <sub>OL1</sub> V <sub>OL2</sub>	0 n/a	0.4 n/a	0 0	0.4 0.6	V
Output fall time: from V <sub>IH</sub> min. to V <sub>IL</sub> max. with a bus capacitance from 10 pF to 400 pF:	t <sub>OF</sub>		250 <sup>2)</sup> n/a	20 + 0.1C <sub>B</sub> <sup>2)</sup> 20 + 0.1C <sub>B</sub> <sup>2)</sup>	250 250 <sup>3)</sup>	ns
with up to 3 mA sink current at V <sub>OL1</sub>		-	250 <sup>2)</sup>	20 + 0.1C <sub>B</sub> <sup>2)</sup>	250	
with up to 6 mA sink current at V <sub>OL2</sub>		n/a	n/a	20 + 0.1C <sub>B</sub> <sup>2)</sup>	250 <sup>3)</sup>	
Input current each I/O pin with an input voltage between 0.4 V and 0.9V <sub>DD</sub> max.	I <sub>i</sub>	-10	10	\$10 <sup>3)</sup>	10 <sup>3)</sup>	μA
Capacitance for each I/O pin	C <sub>i</sub>	-	10	-	10	pF

## NOTES:

n/a = not applicable

1. maximum V<sub>IH</sub> = V<sub>DD</sub> max. + 0.5 V2. C<sub>B</sub> = capacitance of one bus line in pF. Note that the maximum t<sub>F</sub> for the SDA and SCL bus lines quoted in Table 4 (300 ns) is longer than the specified maximum t<sub>OF</sub> for the output stages (250 ns). This allows series protection resistors (R<sub>P</sub>) to be connected between the SDA/SCL pins and the SDA/SCL bus as shown in Fig.37 without exceeding the maximum specified t<sub>F</sub>.3. I/O pins of fast-mode devices must not obstruct the SDA and SCL lines if V<sub>DD</sub> is switched off.

## 14.0 GENERAL CALL ADDRESS AND START BYTE

The 10-bit addressing procedure for the I<sup>2</sup>C-bus is such that the first two bytes after the START condition (S) usually determine which slave will be selected by the master. The exception is the 'general call' address 00000000 (H'00'). Slave devices with 10-bit addressing will react to a 'general call' in the same way as slave devices with 7-bit addressing (see Section 9.1.1).

Hardware masters can transmit their 10-bit address after a 'general call'. In this case, the 'general call' address byte is followed by two successive bytes containing the 10-bit address of the master-transmitter. The format is as shown in Fig.17 where the first DATA byte contains the eight least-significant bits of the master address.

The START byte 00000001 (H'01') can precede the 10-bit addressing in the same way as for 7-bit addressing (see Section 9.1.2).

## 15.0 ELECTRICAL SPECIFICATIONS

The I/O levels, I/O current, spike suppression, output slope control and pin capacitance for I<sup>2</sup>C-bus devices are given in Table 3. The I<sup>2</sup>C-bus timing is given in Table 4. Figure 34 shows the timing definitions for the I<sup>2</sup>C-bus.

The noise margin for HIGH and LOW levels on the bus lines for fast-mode devices are the same as those specified in Section 10.0 for standard-mode I<sup>2</sup>C-bus devices.

The minimum HIGH and LOW periods of the SCL clock specified in Table 4 determine the maximum bit transfer rates of 100 kbit/s for standard-mode devices and 400 kbit/s for fast mode devices. Standard-mode and fast-mode I<sup>2</sup>C-bus devices must be able to follow transfers at their own maximum bit rates, either by being able to transmit or receive at that speed or by applying the clock synchronization procedure described in Section 7 which will force the master into a wait state and stretch the LOW period of the SCL signal. Of course, in the latter case the bit transfer rate is reduced.

## 16.0 APPLICATION INFORMATION

### 16.1 Slope-Controlled Output Stages of Fast-Mode I<sup>2</sup>C-Bus Devices

The electrical specifications for the I/Os of

I<sup>2</sup>C-bus devices and the characteristics of the bus lines connected to them are given in Tables 3 and 4 in Section 15.

Figures 35 and 36 show examples of output stages with slope control in CMOS and bipolar technology. The slope of the falling edge is defined by a Miller capacitor (C1) and a resistor (R1). The typical values for C1 and R1 are indicated on the diagrams. The wide tolerance for output fall time  $t_{OF}$  given in Table 3 means that the design is not critical. The fall time is only slightly influenced by the external bus load ( $C_b$ ) and external pull-up resistor ( $R_p$ ). However, the rise time ( $t_R$ ) specified in Table 4 is mainly determined by the bus load capacitance and the value of the pull-up resistor.

### 16.2 Switched Pull-Up Circuit for Fast-Mode I<sup>2</sup>C-Bus Devices

The supply voltage ( $V_{DD}$ ) and the maximum output LOW level determine the minimum value of pull-up resistor  $R_p$  (see Section 10.1). For example, with a supply voltage of  $V_{DD} = 5V \pm 10\%$  and  $V_{OLmax} = 0.4V$  at 3 mA,  $R_{pmin} = (5.5 - 0.4)/0.003 = 1.7 k\Omega$ . As shown in Fig.38, this value of  $R_p$  limits the maximum bus capacitance to about 200 pF to meet the maximum  $t_R$  requirement of 300 ns. If the bus has a higher capacitance than this, a switched pull-up circuit as shown in Fig.37 can be used.

The switched pull-up circuit in Fig.37 is for a supply voltage of  $V_{DD} = 5V \pm 10\%$  and a maximum capacitive load of 400 pF. Since it is controlled by the bus levels, it needs no additional switching control signals. During the rising/falling edges, the bilateral switch in the HCT4066 switches pull-up resistor  $R_{p2}$  on/off at bus levels between 0.8 V and 2.0 V. Combined resistors  $R_{p1}$  and  $R_{p2}$  can pull-up the bus line within the maximum specified rise time ( $t_R$ ) of 300 ns. The maximum sink current for the driving I<sup>2</sup>C-bus device will not exceed 6 mA at  $V_{OL2} = 0.6V$ , or 3 mA at  $V_{OL1} = 0.4V$ .

Series resistors  $R_s$  are optional. They protect the I/O stages of the I<sup>2</sup>C-bus devices from high-voltage spikes on the bus lines, and minimize crosstalk and undershoot of the bus line signals. The maximum value of  $R_s$  is determined by the maximum permitted voltage drop across this resistor when the bus line is switched to the LOW level in order to switch off  $R_{p2}$ .

### 16.3 Wiring Pattern of the Bus Lines

In general, the wiring must be so chosen that crosstalk and interference to/from the bus lines is minimized. The bus lines are most susceptible to crosstalk and interference at the HIGH level because of the relatively high impedance of the pull-up devices.

If the length of the bus lines on a PCB or ribbon cable exceeds 10 cm and includes the  $V_{DD}$  and  $V_{SS}$  lines, the wiring pattern must be:

SDA \_\_\_\_\_  
 $V_{DD}$  \_\_\_\_\_  
 $V_{SS}$  \_\_\_\_\_  
SCL \_\_\_\_\_

If only the  $V_{SS}$  line is included, the wiring pattern must be:

SDA \_\_\_\_\_  
 $V_{SS}$  \_\_\_\_\_  
SCL \_\_\_\_\_

These wiring patterns also result in identical capacitive loads for the SDA and SCL lines. The  $V_{SS}$  and  $V_{DD}$  lines can be omitted if a PCB with a  $V_{SS}$  and/or  $V_{DD}$  layer is used.

If the bus lines are twisted-pairs, each bus line must be twisted with a  $V_{SS}$  return. Alternatively, the SCL line can be twisted with a  $V_{SS}$  return, and the SDA line twisted with a  $V_{DD}$  return. In the latter case, capacitors must be used to decouple the  $V_{DD}$  line to the  $V_{SS}$  line at both ends of the twisted pairs.

If the bus lines are shielded (shield connected to  $V_{SS}$ ), interference will be minimized. However, the shielded cable must have low capacitive coupling between the SDA and SCL lines to minimize crosstalk.

### 16.4 Maximum and Minimum Values of Resistors $R_p$ and $R_s$ for Fast-Mode I<sup>2</sup>C-Bus Devices

The maximum and minimum values for resistors  $R_p$  and  $R_s$  connected to a fast-mode I<sup>2</sup>C-bus can be determined from Fig.25, 26 and 28 in Section 10.1. Because a fast-mode I<sup>2</sup>C-bus has faster rise times ( $t_R$ ) the maximum value of  $R_p$  as a function of bus capacitance is less than that shown in Fig.27. The replacement graph for Fig.27 showing the maximum value of  $R_p$  as a function of bus capacitance ( $C_b$ ) for a fast mode I<sup>2</sup>C-bus is given in Fig.38.

The I<sup>2</sup>C-bus and how to use itThe I<sup>2</sup>C-bus and how to use itTable 4. Characteristics of the SDA and SCL Bus Lines for I<sup>2</sup>C-Bus Devices

Parameter	Symbol	Standard-mode I <sup>2</sup> C-bus		Fast-mode I <sup>2</sup> C-bus		Unit
		Min.	Max.	Min.	Max.	
SCL clock frequency	f <sub>SCL</sub>	0	100	0	400	kHz
Bus free time between a STOP and START condition	t <sub>BUF</sub>	4.7	-	1.3	-	μs
Hold time (repeated) START condition. After this period, the first clock pulse is generated	t <sub>HD;STA</sub>	4.0	-	0.6	-	μs
LOW period of the SCL clock	t <sub>LOW</sub>	4.7	-	1.3	-	μs
HIGH period of the SCL clock	t <sub>HIGH</sub>	4.0	-	0.6	-	μs
Set-up time for a repeated START condition	t <sub>SU;STA</sub>	4.7	-	0.6	-	μs
Data hold time: for CBUS compatible masters (see NOTE, Section 9.1.3) for I <sup>2</sup> C-bus devices	t <sub>HD;DAT</sub>	5.0	-	-	-	μs
		0 <sup>1)</sup>	-	0 <sup>1)</sup>	0.9 <sup>2)</sup>	μs
Data set-up time	t <sub>SU;DAT</sub>	250	-	100 <sup>3)</sup>	-	ns
Rise time of both SDA and SCL signals	t <sub>R</sub>	-	1000	20 + 0.1C <sub>b</sub> <sup>4)</sup>	300	ns
Fall time of both SDA and SCL signals	t <sub>F</sub>	-	300	20 + 0.1C <sub>b</sub> <sup>4)</sup>	300	ns
Set-up time for STOP condition	t <sub>SU;STO</sub>	4.0	-	0.6	-	μs
Capacitive load for each bus line	C <sub>b</sub>	-	400	-	400	pF

## NOTES:

All values referred to V<sub>IH min.</sub> and V<sub>IL max.</sub> levels (see Table 3).

1. A device must internally provide a hold time of at least 300 ns for the SDA signal (referred to the V<sub>IH min.</sub> of the SCL signal) in order to bridge the undefined region of the falling edge of SCL.
2. The maximum t<sub>HD;DAT</sub> has only to be met if the device does not stretch the LOW period (t<sub>LOW</sub>) of the SCL signal.
3. A fast-mode I<sup>2</sup>C-bus device can be used in a standard-mode I<sup>2</sup>C-bus system, but the requirement t<sub>SU;DAT</sub> ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line t<sub>R max.</sub> + t<sub>SU;DAT</sub> = 1000 + 250 = 1250 ns (according to the standard-mode I<sup>2</sup>C-bus specification) before the SCL line is released.
4. C<sub>b</sub> = total capacitance of one bus line in pF.

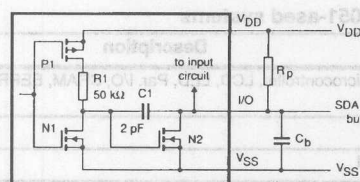


Figure 35. Slope-Controlled Output Stage in CMOS Technology

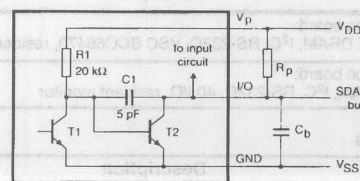
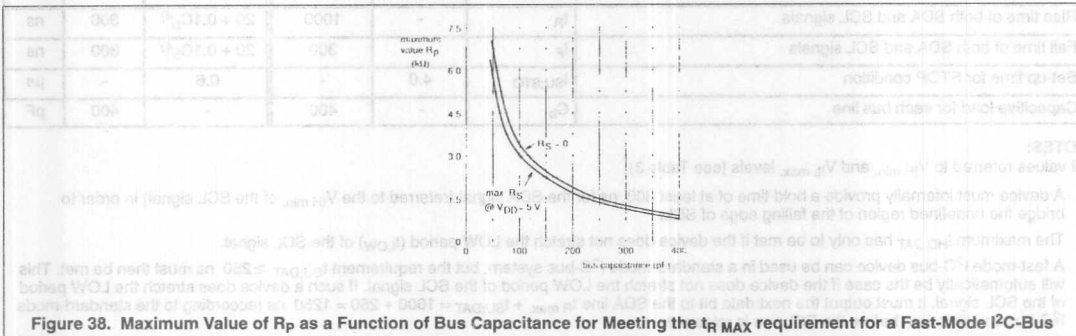
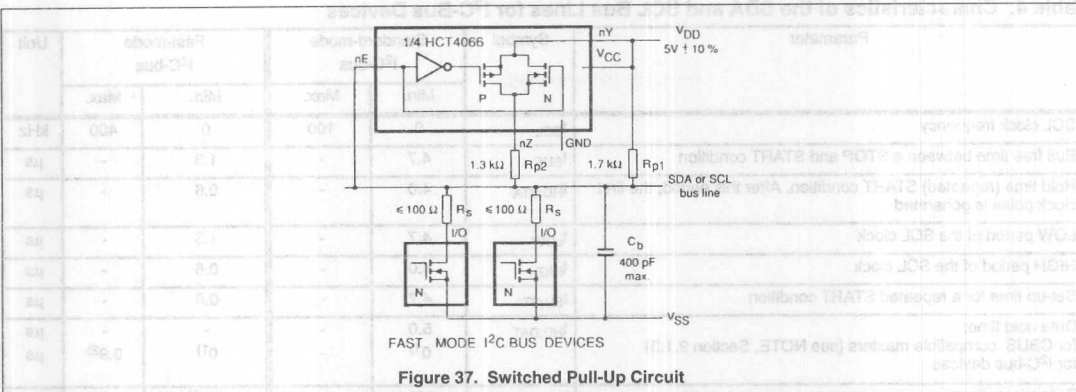


Figure 36. Slope-Controlled Output Stage in Bipolar Technology

# The I<sup>2</sup>C-bus and how to use it



## 17.0 DEVELOPMENT TOOLS

### 17.1 Development tools for 8048 and 8051-based systems

Product	Description
OM1016	I <sup>2</sup> C-bus demonstration board with microcontroller, LCD, LED, Par. I/O, SRAM, EEPROM, Clock, DTMF generator, AD/DA conversion, infrared link.
OM1018	manual for OM1016
OM1020	LCD and driver demonstration board
OM4151	I <sup>2</sup> C-bus evaluation board (similar to OM1016 above but without infrared link).

### 17.2 Development tools for 68000-based systems

Product	Description
OM4160	Microcore-1 demonstration/evaluation board: SCC68070, 128K EPROM, 512K DRAM, I <sup>2</sup> C, RS-232C, VSC SCC66470, resident monitor
OM4160/3	Microcore-3 demonstration/evaluation board: 93C110, 128K EPROM, 64K SRAM, I <sup>2</sup> C, RS-232C, 40 I/O, resident monitor

### 17.3 Development tools for all systems

Product	Description
OM1022	I <sup>2</sup> C-bus analyzer. Hardware and software (runs on IBM or compatible PC) to experiment with and analyze the behaviour of the I <sup>2</sup> C-bus (includes documentation)

## The I<sup>2</sup>C-bus and how to use it

### 18.0 SUPPORT LITERATURE

Data handbooks	
IC01 1992:	Semiconductors for radio and audio systems
IC02 1992:	Semiconductors for television and video systems
IC03 1993:	Semiconductors for telecom systems
IC14 1992:	8048-based 8-bit microcontrollers
IC20 1994:	8051-based 8-bit microcontrollers
Brochures/leaflets	
Microcontrollers and microprocessors for embedded control applications	
I <sup>2</sup> C-bus compatible ICs and support overview	
I <sup>2</sup> C-bus control programs for consumer applications	

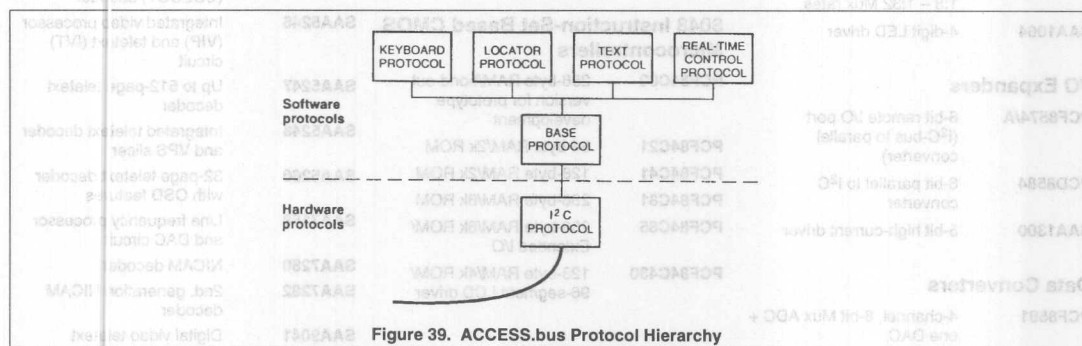


Figure 39. ACCESS.bus Protocol Hierarchy

### 19.0 APPLICATION OF THE I<sup>2</sup>C-BUS IN THE ACCESS.bus SYSTEM

The ACCESS.bus (bus for connecting ACCESSory devices to a host system) is an I<sup>2</sup>C-bus based open-standard serial interconnect system jointly developed and defined by Philips Semiconductors and Digital Equipment Corporation. It is a lower-cost alternative to an RS-232C interface for connecting up to 14 inputs/outputs from peripheral equipment to a desk-top computer or workstation over a distance of up to eight metres. The peripheral equipment can be relatively low speed items such as keyboards, hand-held image scanners, cursor positioners, bar-code readers, digitizing tablets, card readers or modems.

All that's required to implement an ACCESS.bus is an 8051-family

microcontroller with an I<sup>2</sup>C-bus interface, and a 4-wire cable carrying a serial data (SDA) line, a serial clock (SCL) line, a ground wire and a 12 V supply line (500 mA max.) for powering the peripherals.

Important features of the ACCESS.bus are that the bit rate is only about 20% less than the maximum bit rate of the I<sup>2</sup>C-bus, and the peripherals don't need separate device drivers. Also, the protocol allows the peripherals to be changed by 'hot-plugging' without re-booting.

As shown in Fig.39, the ACCESS.bus protocol comprises three levels: the I<sup>2</sup>C-bus protocol, the base protocol, and the application protocol.

The base protocol is common to all ACCESS.bus devices and defines the format of the ACCESS.bus message. Unlike the I<sup>2</sup>C-bus protocol, it restricts masters to sending and slaves to receiving data. One

item of appended information is a checksum for reliability control. The base protocol also specifies seven types of control and status messages which are used in the system configuration which assigns unique addresses to the peripherals without the need for setting jumpers or switches on the devices.

The application protocol defines the message semantics that are specific to the three categories of peripheral device (keyboards, cursor locators, and text devices which generate character streams e.g. card readers) which are at present envisaged.

Philips Semiconductors offers computer peripheral equipment manufacturers technical support, a wide range of I<sup>2</sup>C-bus devices and development kits for the ACCESS.bus. Hardware, software and marketing support is also offered by DEC.



I<sup>2</sup>C peripheral selection guide

## GENERAL PURPOSE ICs

## LCD Drivers

PCF8566	96-segment LCD driver 1:1 – 1:4 Mux rates
PCF8568	LCD row driver for dot matrix displays
PCF8569	LCD Column driver for dot matrix displays
PCF8576	160-segment LCD driver 1:1 – 1:4 Mux rates
PCF8577C	64-segment LCD driver 1:1 – 1:2 Mux Rates
PCF8578/79	Row/column LCD dot-matrix driver/display 1:8 – 1:32 Mux rates

SAA1064	4-digit LED driver
---------	--------------------

## I/O Expanders

PCF8574/A	8-bit remote I/O port (I <sup>2</sup> C-bus to parallel converter)
PCD8584	8-bit parallel to I <sup>2</sup> C converter
SAA1300	5-bit high-current driver

## Data Converters

PCF8591	4-channel, 8-bit Mux ADC + one DAC
TDA8442	Quad 6-bit DAC
TDA8444	Octal 6-bit DAC

## Memory

PCF8570/C	256-byte static RAM
PCF8571	128-byte static RAM
PCF8581	128-byte EEPROM
PC.8582	256-byte EEPROM
PCF8583	256-byte RAM/clock/calendar
PCF8594	512-byte EEPROM
PCF8598	1K-byte EEPROM

## Clocks/Calendars

PCF8573	Clock/calendar
PCF8583	Clock/calendar/ 256-byte RAM

68000-Based CMOS  
Microcontrollers

68070	68000 CPU/MMU/UART/ DMA/timer
93CXXX	UST/I <sup>2</sup> C/34k ROM/ 512 RAM

80C51-Based CMOS  
Microcontrollers\*

8XCL410	4k ROM/128 RAM, low power
8XC528	32k ROM/512 RAM, T2, WD
8XC552	256-byte RAM/8k ROM/ ADC/UART/PWM
8XC652	256-byte RAM/8k ROM, UART
8XC654	256-byte RAM/16kROM, UART
8XC751	64-byte RAM/2k ROM
8XC752	64-byte RAM/2k ROM, ADC/PWM

8048 Instruction-Set Based CMOS  
Microcontrollers

PCF84C00	256-byte RAM/bond-out version for prototype development
PCF84C21	64-byte RAM/2k ROM
PCF84C41	128-byte RAM/2k ROM
PCF84C81	256-byte RAM/8k ROM
PCF84C85	256-byte RAM/8k ROM/ Extended I/O
PCF84C430	128-byte RAM/4k ROM/ 96-segment LCD driver

## MULTIMEDIA ICs

## Desktop Videos

SAA7151B	8-bit digital multistandard TV decoder
SAA7152	Digital comb filter
SAA7157	Clock signal generation circuit for digital video systems; for use with SAA71xx
SAA7165	Video enhancement and D/A processor including digital CTI
SAA7186	Digital video scaler
SAA7191	Digital multistandard TV decoder, square pixel
SAA7191B	SAA7191 variant
SAA7192A	Digital colour space converter with independent LHT
SAA7199B	digital multistandard encoder
SAA9051	Digital multistandard (PAL/NTSC) colour decoder
SAA9056	Digital SECAM colour decoder

## SAA9057B

Clock signal generation  
circuit for digital video  
systems; for use with  
SAA90xx

## SAA9065

Video enhancement and  
D/A processor

## TDA4680

Video processor

## TDA8440

Video/audio switch

## Video/Radio/Audio

## SAA5243

Enhanced Computer  
Controlled Teletext (ECCT)  
decoder

## SAA5244

Single-chip teletext decoder

## SAA5245

Enhanced Computer  
Controlled Teletext  
(USECCT) decoder

## SAA5246

Integrated video processor  
(VIP) and teletext (IVT)  
circuit

## SAA5247

Up to 512-page teletext  
decoder

## SAA5248

Integrated teletext decoder  
and VPS slicer

## SAA5260

32-page teletext decoder  
with OSD features

## SAA7158

Line frequency processor  
and DAC circuit

## SAA7280

NICAM decoder

## SAA7282

2nd. generation NICAM  
decoder

## SAA9041

Digital video teletext  
(DVTB) processor

## SAA9042

Digital video teletext  
(DVTB) processor

## SAB3035/36/37

Digital tuning circuits for  
computer-controlled TV

## SAB9070

Picture-in-picture processor

## SAF1135

Dataline 16 decoder for  
VCR

## TDA1551Q

2 X 22W BTL audio power  
amp with diagnostic

## TDA4670

Picture signal improvement  
circuit

## TDA4671

Picture signal improvement  
circuit

## TDA4681

Video processor with  
automatic cut-off and white  
level control

## TDA4685

Video processor

## TDA4686

Video processor (100 Hz)

## TDA4687

Video processor

## TDA8415/17

Stereo/dual sound  
processor

## TDA8421

Audio processor with a  
loudspeaker channel and a  
headphone channel

## TDA8425

Audio processor with a  
loudspeaker channel only

## TDA8426

Hi-fi stereo audio processor

I<sup>2</sup>C peripheral selection guide

## Video/Radio/Audio (Continued)

<b>TDA8540</b>	4x4 video switch matrix
<b>TDA9140</b>	Alignment-free multistandard decoder
<b>TEA6320</b>	4 input Tone/volume controller with fader control
<b>TEA6330</b>	Tone/volume controller
<b>TSA6060</b>	A/M Frequency Synthesizer for RDS.
<b>SAA9041</b>	Digital video teletext (DVTB) processor
<b>SAB3035/36/7</b>	Digital tuning circuits for computer-controlled TV
<b>SAF1135</b>	Dataline 16 decoder for VCR
<b>TDA1551Q</b>	2 x 22W Audio Power Amp
<b>TDA4670</b>	Picture signal improvement circuit
<b>TDA6360</b>	5 Band Equalizer
<b>TDA8415/17</b>	Stereo/dual sound processor
<b>TDA8421</b>	Audio processor with a loudspeaker channel and a headphone channel

<b>TDA8425</b>	Audio processor with a loudspeaker channel only
<b>TDA8433</b>	Deflection processor
<b>TDA8442</b>	Interface for color decoders
<b>TDA8443/A</b>	YUV/RGB matrix switch
<b>TDA8461</b>	PAL/NTSC color decoder and RGB processor
<b>TDA8466</b>	PAL/NTSC color decoder and RGB processor
<b>TDA9150</b>	Deflection processor
<b>TEA6100</b>	FM/IF and digital tuning IC for computer-controlled radio
<b>TEA6300</b>	Sound fader control and preamplifier/source selector for car radio
<b>TEA6310T</b>	Sound fader control with tone and volume control for car radio
<b>TSA5511/12/14</b>	PLL frequency synthesizer for TV
<b>TSA6057</b>	PLL frequency synthesizer for radio

## Telecom

<b>NE5750/51</b>	Audio processor pair
<b>NE5752</b>	3 V 5750 variant (samples Q4 92)
<b>NE5753</b>	3 V 5751 variant (samples Q4 92)
<b>PCD3311/12</b>	Tone generator (DTMF/modem/musical)
<b>PCD3341</b>	Advanced 10 to 110-number repertory dialer with LCD control
<b>PCD3343</b>	Microcontroller with 224-byte RAM/3k ROM
<b>PCD3348</b>	Microcontroller with 256-byte RAM/8k ROM
<b>UMA1000T</b>	Data processor for mobile telephones
<b>UMA1014T</b>	1GHz frequency synthesizer for mobile telephones
<b>UMF1009</b>	Frequency synthesizer

\* Also available with extended temperature ranges.

FOR FURTHER INFORMATION ON THESE DEVICES, REFER TO *I<sup>2</sup>C-PERIPHERALS FOR MICROCONTROLLERS DATA HANDBOOK*, AVAILABLE FROM YOUR LOCAL PHILIPS SEMICONDUCTORS SALES OFFICE (SEE SECTION 10 OF THIS BOOK).

## QUICK REFERENCE DATA

SYMBOL	PARAMETER	LIMITS			UNIT
		MIN.	TYP.	MAX.	
V <sub>CC</sub>	Supply voltage	4.5		12	V
I <sub>CC</sub>	Quiescent current		18		mA
I <sub>OL</sub>	Output sink capability	30			mA
V <sub>IL</sub>	Input voltage range	0		V <sub>CC</sub>	V
V <sub>OL</sub>	Output voltage range	0		V <sub>CC</sub>	V
ΔT <sub>st</sub>	Maximum temperature rise	8	10	13	°C
T <sub>amb</sub>	Temperature range	-40		+85	°C

## ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
8-Pin Plastic Dual in-line (DIP) Package	P82B718P-N	90T87
8-Pin Plastic SOL (Small Outline Large) Dual in-line (DIP) Package	P82B718T-D	90T88

82B715 is available in chip form

I<sup>2</sup>C bus extender

82B715

DESCRIPTION

The 82B715 is a bipolar integrated circuit intended for application in I<sup>2</sup>C bus systems.

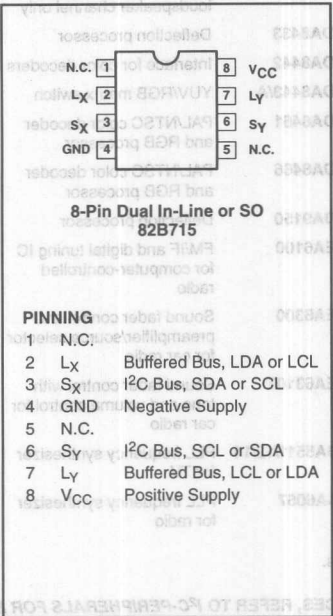
While retaining all the operating modes and features of the I<sup>2</sup>C system it permits extension of the practical separation distance between components on the I<sup>2</sup>C bus by buffering both the data (SDA) and the clock (SCL) lines.

The I<sup>2</sup>C bus capacitance limit of 400pF restricts practical communication distances to a few meters. Using one 82B715 at each end of longer cables reduces the cable loading capacitance on the I<sup>2</sup>C bus by a factor of 10, times and may allow the use of low cost general purpose wiring to extend bus lengths.

**FEATURES**

- Dual, bi-directional, unity voltage gain buffer
- I<sup>2</sup>C bus compatible
- Logic signal levels may include both supply and ground
- X10 impedance transformation
- Wide supply voltage range

PIN CONFIGURATIONS



QUICK REFERENCE DATA

SYMBOL	PARAMETER	LIMITS			UNIT
		MIN.	TYP.	MAX.	
V <sub>CC</sub>	Supply voltage	4.5		12	V
I <sub>CC</sub>	Quiescent current		16		mA
I <sub>line</sub>	Output sink capability	30			mA
V <sub>in</sub>	Input voltage range	0		V <sub>CC</sub>	V
V <sub>out</sub>	Output voltage range	0		V <sub>CC</sub>	V
Z <sub>in</sub> /Z <sub>out</sub>	Impedance transformation	8	10	13	
T <sub>amb</sub>	Temperature range	−40		+85	°C

ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
8-Pin Plastic Dual In-Line (N/P) Package	P82B715P N	SOT97
8-Pin Plastic SOL (Small Outline Large) Dual In-Line (D/T) Package	P82B715T D	SOT96A
82B715 is available in chip form		

I<sup>2</sup>C bus extender

82B715

FUNCTIONAL DESCRIPTION

The 82B715 bipolar integrated circuit contains two identical buffer circuits which enable I<sup>2</sup>C and similar bus systems to be extended over long distances without degradation of system performance or requiring the use of special cables.

The buffer has an effective current gain of ten from I<sup>2</sup>C bus to Buffered bus. Whatever current is flowing out of the I<sup>2</sup>C bus side, ten times that current will be flowing into the Buffered bus side (see Figure 2).

As a consequence of this amplification the system is able to drive capacitive loads up to ten times the standard limit on the Buffered bus side. This current based buffering

approach preserves the bi-directional, open-collector/open-drain characteristic of the I<sup>2</sup>C SDA/SCL lines.

To minimize interference and ensure stability, current rise and fall rates are internally controlled.

APPLICATION NOTES

By using two (or more) 82B715 ICs, a sub-system can be built which retains the interface characteristics of an I<sup>2</sup>C device so that it may be included in, or optionally added to, any I<sup>2</sup>C or related system.

The sub-system features a low impedance or "Buffered" bus, capable of driving large wiring capacities (see Figure 3).

I<sup>2</sup>C Systems

As with the standard I<sup>2</sup>C system, pull-up resistors are required to provide the logic HIGH levels on the Buffered bus. (Standard open-collector configuration of the I<sup>2</sup>C bus). The size and number of these pull-up resistors depends on the system.

If the buffer is to be permanently connected into the system, the circuit should be configured with only one pull-up resistor on the Buffered bus and none on the I<sup>2</sup>C bus.

Alternatively a buffer may be connected to an existing I<sup>2</sup>C system. In this case the Buffered bus pull-up will act in parallel with the I<sup>2</sup>C bus pull-up.

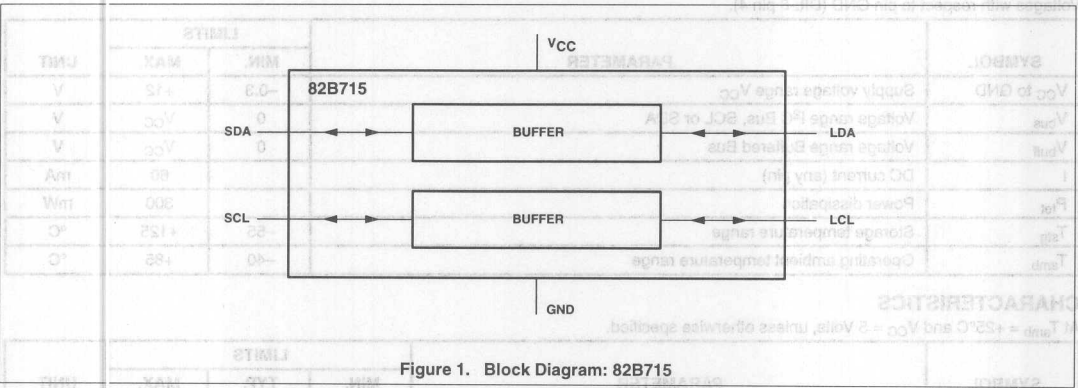


Figure 1. Block Diagram: 82B715

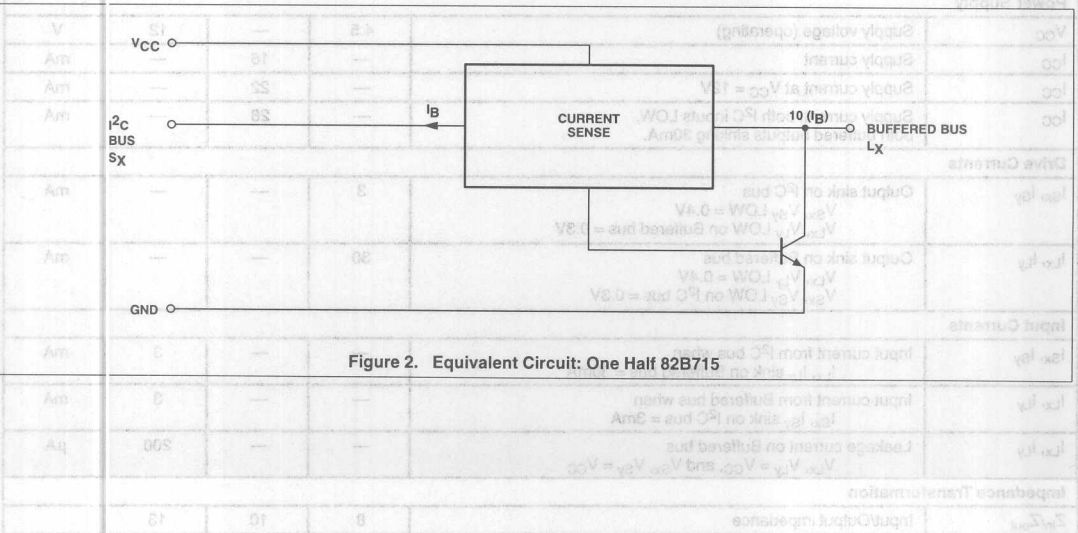


Figure 2. Equivalent Circuit: One Half 82B715

I<sup>2</sup>C bus extender

82B715

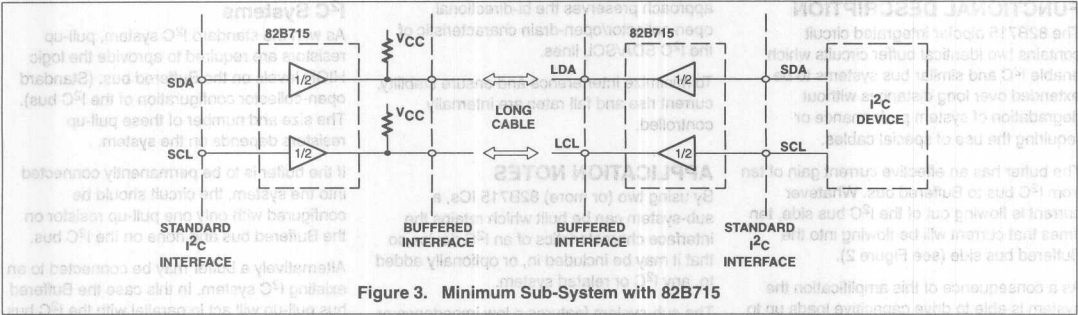


Figure 3. Minimum Sub-System with 82B715

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134).  
Voltages with respect to pin GND (DIL-8 pin 4).

SYMBOL	PARAMETER	LIMITS		UNIT
		MIN.	MAX.	
V <sub>CC</sub> to GND	Supply voltage range V <sub>CC</sub>	-0.3	+12	V
V <sub>bus</sub>	Voltage range I <sup>2</sup> C Bus, SCL or SDA	0	V <sub>CC</sub>	V
V <sub>buff</sub>	Voltage range Buffered Bus	0	V <sub>CC</sub>	V
I	DC current (any pin)		60	mA
P <sub>tot</sub>	Power dissipation		300	mW
T <sub>stg</sub>	Storage temperature range	-55	+125	°C
T <sub>amb</sub>	Operating ambient temperature range	-40	+85	°C

CHARACTERISTICS

At T<sub>amb</sub> = +25°C and V<sub>CC</sub> = 5 Volts, unless otherwise specified.

SYMBOL	PARAMETER	LIMITS			UNIT
		MIN.	TYP.	MAX.	
Power Supply					
V <sub>CC</sub>	Supply voltage (operating)	4.5	—	12	V
I <sub>CC</sub>	Supply current	—	16	—	mA
I <sub>CC</sub>	Supply current at V <sub>CC</sub> = 12V	—	22	—	mA
I <sub>CC</sub>	Supply current; both I <sup>2</sup> C inputs LOW, both buffered outputs sinking 30mA.	—	28	—	mA
Drive Currents					
I <sub>Sx</sub> , I <sub>Sy</sub>	Output sink on I <sup>2</sup> C bus V <sub>Sx</sub> , V <sub>Sy</sub> LOW = 0.4V V <sub>Lx</sub> , V <sub>Ly</sub> LOW on Buffered bus = 0.3V	3	—	—	mA
I <sub>Lx</sub> , I <sub>Ly</sub>	Output sink on Buffered bus V <sub>Lx</sub> , V <sub>Ly</sub> LOW = 0.4V V <sub>Sx</sub> , V <sub>Sy</sub> LOW on I <sup>2</sup> C bus = 0.3V	30	—	—	mA
Input Currents					
I <sub>Sx</sub> , I <sub>Sy</sub>	Input current from I <sup>2</sup> C bus when I <sub>Lx</sub> , I <sub>Ly</sub> sink on Buffered bus = 30mA	—	—	3	mA
I <sub>Lx</sub> , I <sub>Ly</sub>	Input current from Buffered bus when I <sub>Sx</sub> , I <sub>Sy</sub> sink on I <sup>2</sup> C bus = 3mA	—	—	3	mA
I <sub>Lx</sub> , I <sub>Ly</sub>	Leakage current on Buffered bus V <sub>Lx</sub> , V <sub>Ly</sub> = V <sub>CC</sub> , and V <sub>Sx</sub> , V <sub>Sy</sub> = V <sub>CC</sub>	—	—	200	μA
Impedance Transformation					
Z <sub>in</sub> /Z <sub>out</sub>	Input/Output impedance	8	10	13	



I<sup>2</sup>C bus extender

## 82B715

**Pull-Up Resistance Calculation**

In calculating the pull-up resistance values, the gain of the buffer introduces scaling factors which must be applied to the system components. Viewing the system from the Buffered bus, all I<sup>2</sup>C bus capacitances have effectively 10 times their I<sup>2</sup>C bus value.

In practical systems the pull-up resistance is determined by the rise time limit for I<sup>2</sup>C systems. As an approximation this limit will be satisfied if the time constant (product of the net resistance and net capacitance) of the total system is set to 1 microsecond.

The total time constant may either be set by considering each bus node individually (i.e., the I<sup>2</sup>C nodes, and the Buffered bus node) and choosing pull-up resistors to give time constants of 1 microsecond for each node; or by combining the capacitances into an equivalent capacitive loading on the Buffered bus, and calculating the Buffered bus pull-up resistor required by this equivalent capacitance.

For each separate bus the pull-up resistor may be calculated as follows:

$$R = \frac{1 \mu \text{sec}}{C_{\text{device}} + C_{\text{wiring}}}$$

Where:  $C_{\text{device}}$  = sum of device capacitances connected to each bus,

and  $C_{\text{wiring}}$  = total wiring and stray capacitance on each bus.

If these capacitances are not known then a good approximation is to assume that each device presents 10pF of load capacitance and 10pF of wiring capacitance.

The capacitance figures for one or more individual I<sup>2</sup>C bus nodes should be multiplied by a factor of 10 times, and then added to the Buffered bus capacitance. Calculation of a new Buffered bus pull-up resistor will allow this single pull-up resistor to act for both the included I<sup>2</sup>C bus nodes and the Buffered bus.

Thus it is possible to combine some or all of these separate pull-up resistors into a single resistor on the Buffered bus (the value of which is calculated from the sum of the scaled capacitances on the Buffered bus). If the buffer is to be permanently connected into the system then all the separate pull-up resistors should be combined. But if it is to be connected by adding it onto an existing system, then only those on the additional I<sup>2</sup>C bus system can be combined onto the Buffered bus if the original system is required to be able to still operate on a stand-alone basis.

A further restriction is that the maximum pull-up current, with the bus LOW, should not exceed the I<sup>2</sup>C bus specification maximum of

3mA, or 30mA on the Buffered bus. The following formula applies:

$$30\text{mA} > \frac{V_{\text{CC}} - 0.4}{R_{\text{P}}}$$

Where:  $R_{\text{P}}$  = scaled parallel combination of all pull-up resistors.

If this condition is met, the fall time specifications will also be met.

Figure 4 shows typical loading calculations for the expanded I<sup>2</sup>C bus.

**Sx, Sy, I<sup>2</sup>C Bus, SDA or SCL**

Because the two buffer circuits in the 82B715 are identical either input pin can be used as the I<sup>2</sup>C Bus SDA data line, or the SCL clock line.

**Lx, Ly, Buffered Bus, LDA or LCL**

On the buffered low impedance line side, the corresponding output becomes LDA and LCL.

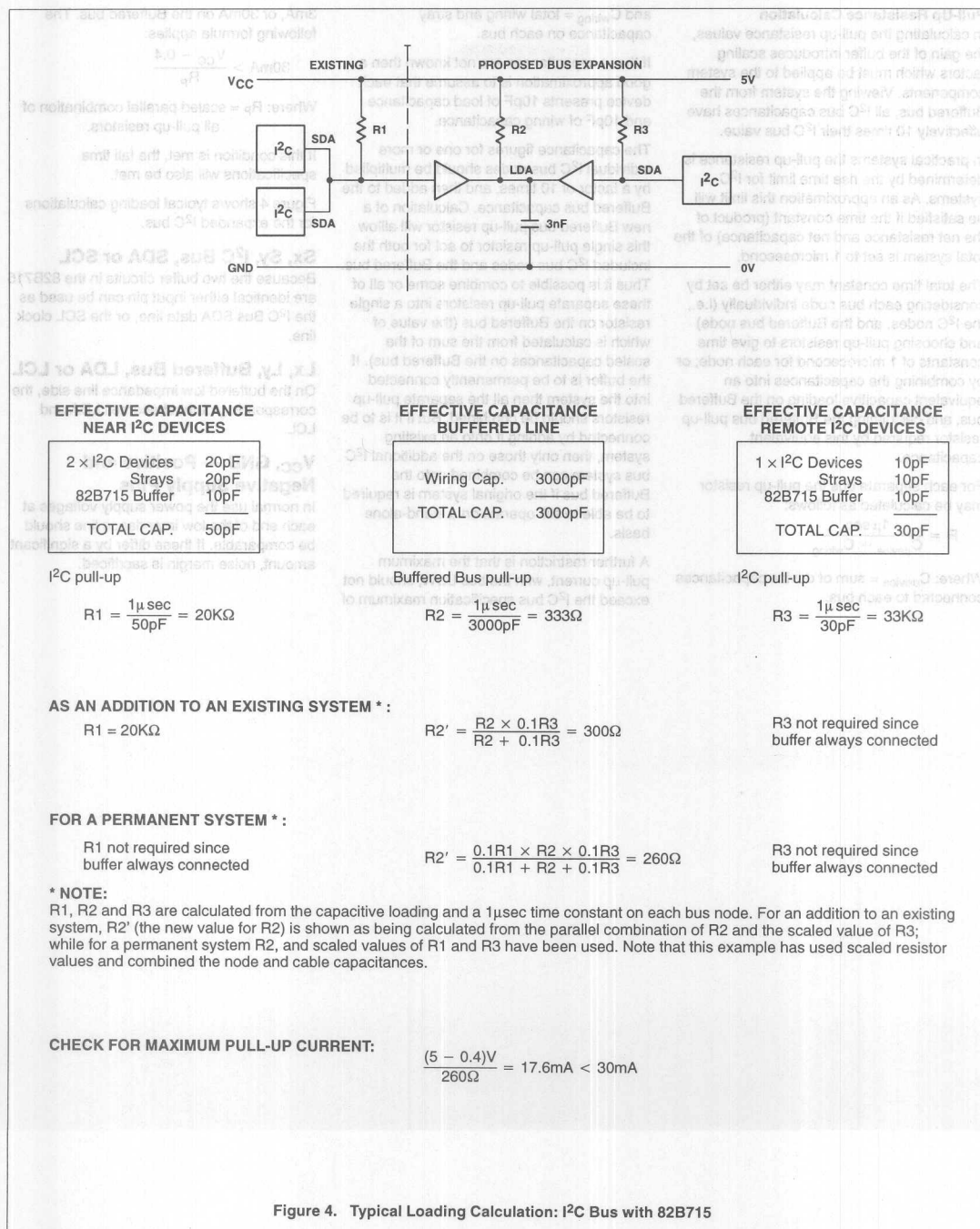
**V<sub>CC</sub>, GND — Positive and Negative Supply Pins**

In normal use the power supply voltages at each end of the low impedance line should be comparable. If these differ by a significant amount, noise margin is sacrificed.

Figure 4. Typical Loading Calculation: I<sup>2</sup>C Bus with 82B715

I<sup>2</sup>C bus extender

## 82B715



# Section 5

## ACCESS.bus Technical Overview

### CONTENTS

<b>ACCESS.bus Technical Overview</b> .....	1169
Introduction .....	1169
What is ACCESS.bus? .....	1169
ACCESS.bus Hardware .....	1169
ACCESS.bus Protocols .....	1170
<b>HOW ACCESS.bus Works</b> .....	1171
Electrical .....	1171
Bus Transactions .....	1171
Synchronization .....	1173
Byte Framing and Acknowledgement .....	1173
Addressing .....	1173
Arbitration .....	1173
Message Format .....	1173
Control/Status Messages .....	1173
Configuration .....	1174
Device Identifiers .....	1175
Device Capabilities Information .....	1175
<b>Application Device Types</b> .....	1176
Keyboard Devices .....	1176
Locator Devices .....	1176
Text Devices .....	1176
<b>Timing Rules</b> .....	1176
Transaction Timing Rules .....	1176
Response Timeouts .....	1176
<b>Software Architecture and Development</b> .....	1176
Device Firmware Development .....	1177
Host Software Architecture .....	1177
<b>Development Support</b> .....	1177
ACCESS.bus Industry Group .....	1177
Philips Semiconductors Support .....	1177
<b>ACCESS.bus development kit</b> .....	1180
<b>ACCESS.bus PC/AT controller board</b> .....	1182



## ACCESS.bus™ technical overview

### INTRODUCTION

ACCESS.bus™ (a BUS for connecting ACCESSory devices to a host system) is a peripheral interconnect system defined and developed by Digital Equipment Corporation and offered to the computer industry as an open standard.

This overview aims to introduce the prospective developer of ACCESS.bus systems or peripherals to the essential technical features of this interconnect. It is meant to be a general technical overview of the ACCESS.bus architecture. Under no circumstances should it be used as the basis for designing any device or system. Developers wishing to design a host system or peripheral device that implements ACCESS.bus should refer to the *ACCESS.bus Hardware and Protocol Specifications Version 2.0*, July 1992, available from the ACCESS.bus Industry Group, Digital's TRI/ADD Program, or Philips. Addresses and other information on support may be found in Section 6.

### What is ACCESS.bus?

ACCESS.bus is a system for connecting a number of relatively low-speed I/O devices to a host computer, typically a desktop system, such as a workstation, personal computer, or terminal. Devices include both interactive

peripherals – keyboards, locators, hand-held scanners, bar code readers, and magnetic card readers – and non-interactive peripherals – printers, and in realtime control applications, signal transducers. Further, the ACCESS.bus protocol is general enough to accommodate a wide range of unusual peripheral types such as data gloves (see Figure 1).

ACCESS.bus has a bus topology architecture. That is, a single ACCESS.bus on a host can accommodate up to 125 peripheral devices. The total length of the cable connecting the devices on a common ACCESS.bus may be up to eight meters. The limiting factors are capacitance, which may not exceed 800pF, and the maximum voltage drop, which must allow maintenance of  $+5V \pm 10\%$ . Using an I<sup>2</sup>C bus extender that maximum distance may be lengthened. ACCESS.bus supports a maximum aggregate data throughput of approximately 80 Kbits/sec.

Digital has made the ACCESS.bus technology an open specification, enabling any vendor to implement it on host systems or in peripheral devices without fee or royalty.

ACCESS.BUS offers a number of advantages both to end users and to the developers of systems and peripheral

devices. A host computer needs only one hardware port to connect to a number of devices. The commonality in communication methods for a number of device types leads to economies in software and hardware development. As an open industry standard, ACCESS.bus will stimulate development of diverse peripheral devices, each usable with a number of different types of host systems.

ACCESS.bus incorporates more sophisticated technology and offers higher performance than any other bus-topology interconnect for desktop peripherals. Moreover, it is the first system of this kind to be offered as an open nonproprietary standard.

### ACCESS.bus Hardware

At the hardware level, ACCESS.bus is based on the well-established Inter-Integrated Circuit (I<sup>2</sup>C) serial bus developed and patented by Philips. The serial bus architecture, in which a single data line carries one bit of information at a time, entails lower costs for cabling, connectors and controller circuitry than parallel bus architectures.

Standard low-cost I<sup>2</sup>C components, available from Philips, handle the logical complications of the bit-level handshaking. More details on these components are given in section 5.

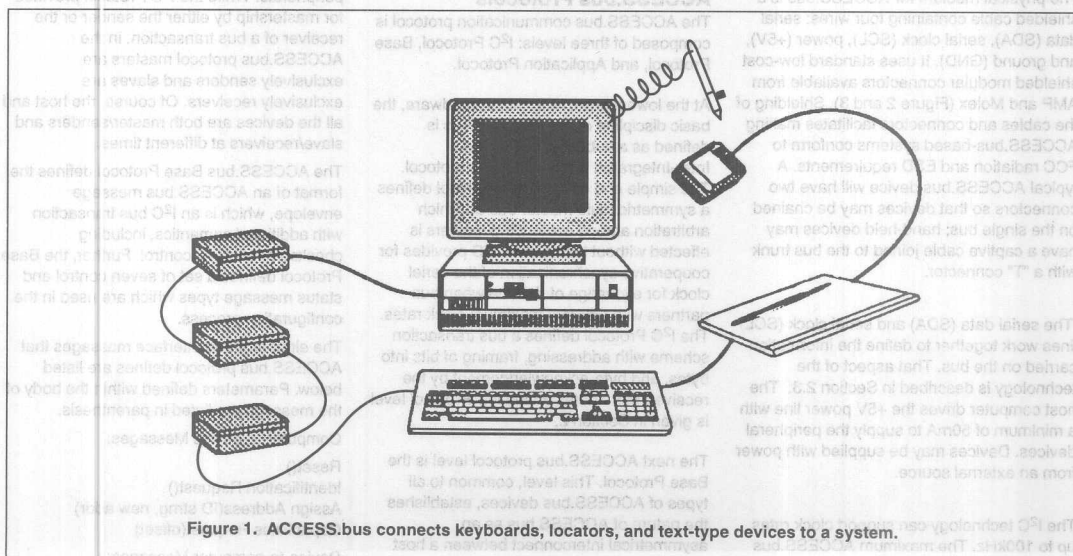


Figure 1. ACCESS.bus connects keyboards, locators, and text-type devices to a system.



## ACCESS.bus™ technical overview

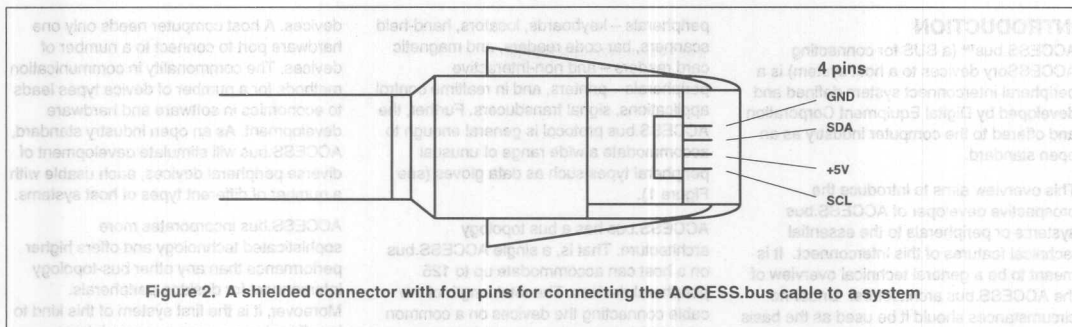


Figure 2. A shielded connector with four pins for connecting the ACCESS.bus cable to a system

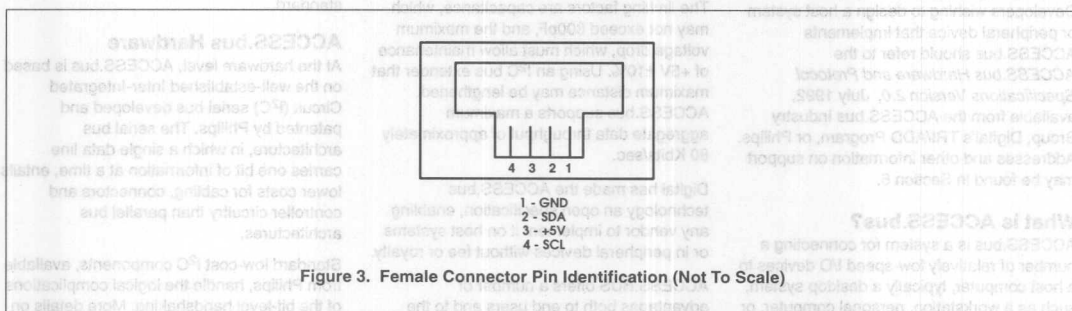


Figure 3. Female Connector Pin Identification (Not To Scale)

The physical medium for ACCESS.bus is a shielded cable containing four wires: serial data (SDA), serial clock (SCL), power (+5V), and ground (GND). It uses standard low-cost shielded modular connectors available from AMP and Molex (Figure 2 and 3). Shielding of the cables and connectors facilitates making ACCESS.bus-based systems conform to FCC radiation and ESD requirements. A typical ACCESS.bus device will have two connectors so that devices may be chained on the single bus; hand-held devices may have a captive cable joined to the bus trunk with a "T" connector.

The serial data (SDA) and serial clock (SCL) lines work together to define the information carried on the bus. That aspect of the technology is described in Section 2.3. The host computer drives the +5V power line with a minimum of 50mA to supply the peripheral devices. Devices may be supplied with power from an external source.

The I<sup>2</sup>C technology can support clock rates up to 100kHz. The maximum ACCESS.bus data transfer rate of approximately 80 Kbits/sec is derived from the top clock rate by subtracting the overhead imposed by the ACCESS.bus communication protocols for handshaking, addressing, and error control.

## ACCESS.bus Protocols

The ACCESS.bus communication protocol is composed of three levels: I<sup>2</sup>C Protocol, Base Protocol, and Application Protocol.

At the lowest level, nearest the hardware, the basic discipline of the ACCESS.bus is defined as a subset of the Philips Inter-Integrated Circuit (I<sup>2</sup>C) bus protocol. The simple and efficient I<sup>2</sup>C Protocol defines a symmetric multi-master bus on which arbitration among contending masters is effected without losing data. I<sup>2</sup>C provides for cooperative synchronization of the serial clock for exchange of data between bus partners with different maximum clock rates. The I<sup>2</sup>C Protocol defines a bus transaction scheme with addressing, framing of bits into bytes, and byte-acknowledgement by the receiver. More detail on the I<sup>2</sup>C Protocol level is given in Section 2.

The next ACCESS.bus protocol level is the Base Protocol. This level, common to all types of ACCESS.bus devices, establishes the nature of ACCESS.bus as an asymmetrical interconnect between a host computer and a number of peripheral devices. The host plays a special role as a manager of the ACCESS.bus. Data Communication is always between host and peripheral device and never between two

peripherals. While the I<sup>2</sup>C Protocol provides for mastership by either the sender or the receiver of a bus transaction, in the ACCESS.bus protocol masters are exclusively senders and slaves are exclusively receivers. Of course, the host and all the devices are both master/senders and slave/receivers at different times.

The ACCESS.bus Base Protocol defines the format of an ACCESS.bus message envelope, which is an I<sup>2</sup>C bus transaction with additional semantics, including checksum reliability control. Further, the Base Protocol defines a set of seven control and status message types which are used in the configuration process.

The eight required interface messages that ACCESS.bus protocol defines are listed below. Parameters defined within the body of the message are listed in parenthesis.

## Computer-to-device Messages:

Reset()  
Identification Request()  
Assign Address(ID string, new addr)  
Capabilities Request(offset)

## Device-to-computer Messages:

Attention(status)  
Identification Reply(ID string)  
Capabilities Reply(offset, data frag)  
Interface Error()

## ACCESS.bus™ technical overview

Two of the unique features of this configuration process are auto-addressing and hot plugging. Auto-addressing refers to the way that devices are assigned unique bus addresses in the configuration process, without the need for setting jumpers or switches on the devices. Hot plugging refers to the ability for attaching or detaching devices while the system is running, without the need for rebooting the host. The means by which the ACCESS.bus protocol provides these features is discussed in Section 2.

The highest level of the ACCESS.bus protocol, the Application Protocol, defines message semantics that are specific to particular functional types of devices. Different device types require different Application Protocols. Application Protocols have been defined so far for three device classes: keyboards, locators, and text devices. Each of these predefined classes is designed to be broad. The keyboard device protocol defines standard messages for reporting keystrokes and controlling keyboard peripherals. The protocol attempts to define the simplest set of functions from which common industry standard keyboard interfaces can be built. The locator device protocol defines a set of standard messages for reporting locator movement and key switch activation for mice, tablets, and other positioning devices. The protocol is designed to accommodate a range of basic locator devices such as a mouse or tablet. More complex devices can be modeled as a combination of basic devices or can provide their own device driver. The text device

protocol is intended to provide a simple way to transmit character or binary data to and from stream oriented devices such as a bar code reader, or modem. The sequential character stream model also serves as a common denominator for connecting RS-232 interface devices.

A major advantage in designing devices that conform to these general device-type semantics is that they may share device-specific software, both in the device-resident firmware and in the driver software needed in the host operating system to allow application programs to access the devices.

It is anticipated that further device-specific Application Protocols will be defined in the future, under the aegis of the ACCESS.bus Industry Group. Further, any device vendor may implement a special device protocol within the general message envelope defined by the Base Protocol.

Participation in all three of the protocol levels requires intelligence at the device level. The lower levels of this firmware are likely to be common to many devices. Higher levels of the firmware are expected to be more specific to the device and the application (Figure 4).

### HOW ACCESS.bus WORKS

#### Electrical

The host and devices are connected to both the serial data (SDA) and serial clock (SCL)

lines in an "wired-AND" logic configuration. The wired-AND may be implemented by connecting the data and clock output stages of each bus partner to the SDA and SCL lines respectively through open-collector or open-drain transistors. The standard I<sup>2</sup>C components include such output stages on-chip. The significance of the wired-AND logic is that any attached bus partner may force either of these lines to low (the ground level). When there is no output from any bus partner, the lines are held high by pull-up current sources in the host. Every bus partner can sense the level on both of these lines (Figure 5).

#### Bus Transactions

During a bus transaction, there is one clock pulse on SCL for each bit transferred on SDA. The SDA information is valid when SCL is high. During a transaction, the SDA must be stable between the rising and falling edges of the SCL pulse; SDA may change state only when SCL is low (Figure 6).

SDA transitions when the SCL is high are signals that delimit the bus transaction. When the ACCESS.bus is free, both SCL and SDA are high. A high-to-low SDA transition when SCL is high is a start condition; it signals the beginning of a bus transaction. A bus partner asserts mastership by pulling SDA low when the bus is free. A low-to-high SDA transition when SCL is high is a stop condition; it signals the end of a bus transaction. A master generates a stop condition when it relinquishes mastership (Figure 7).

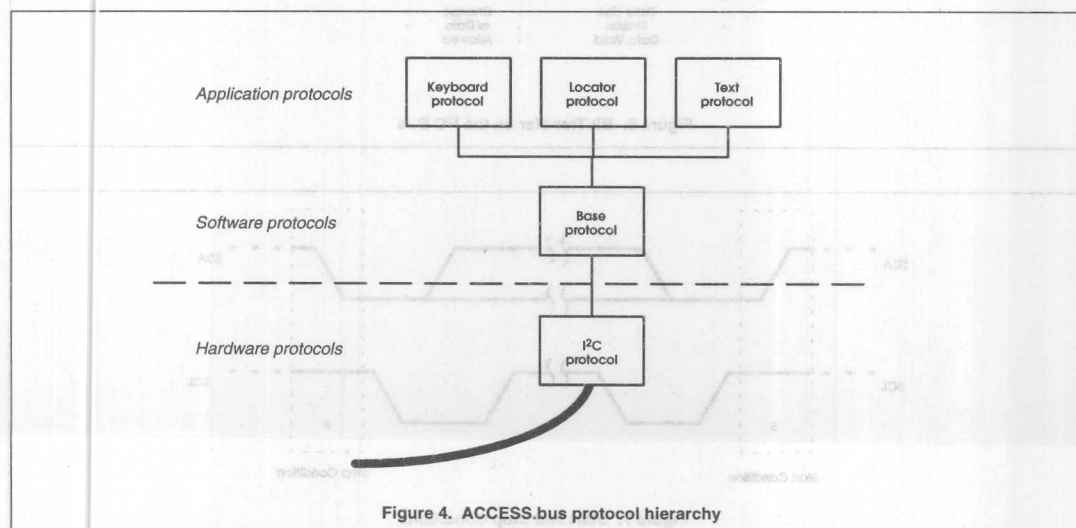
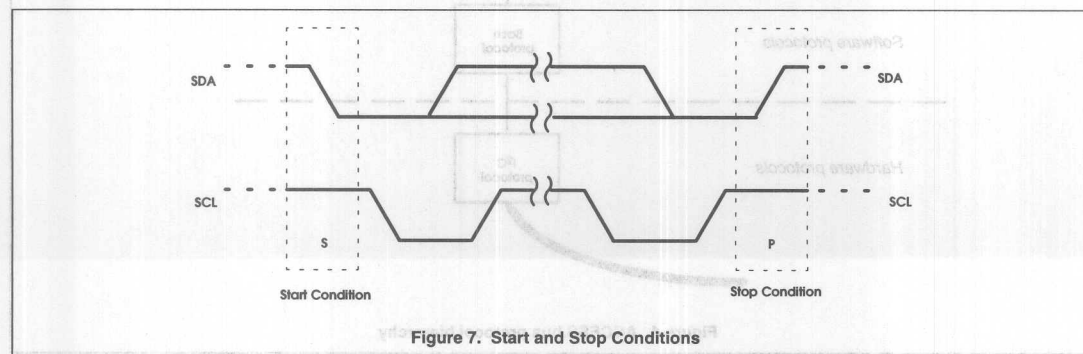
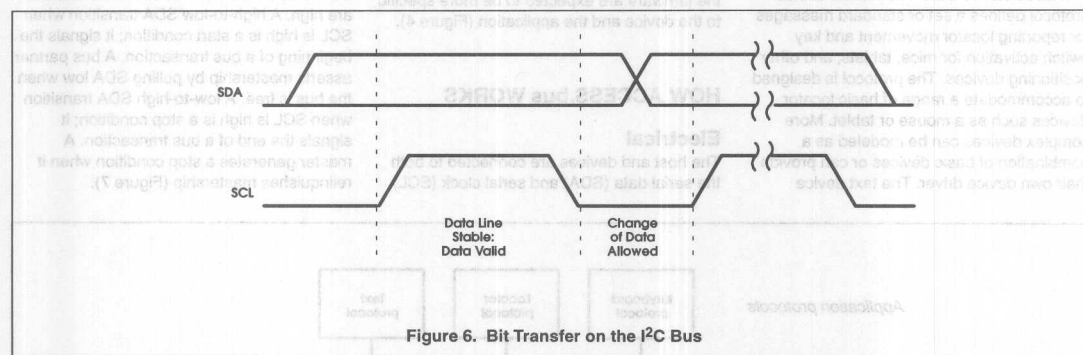
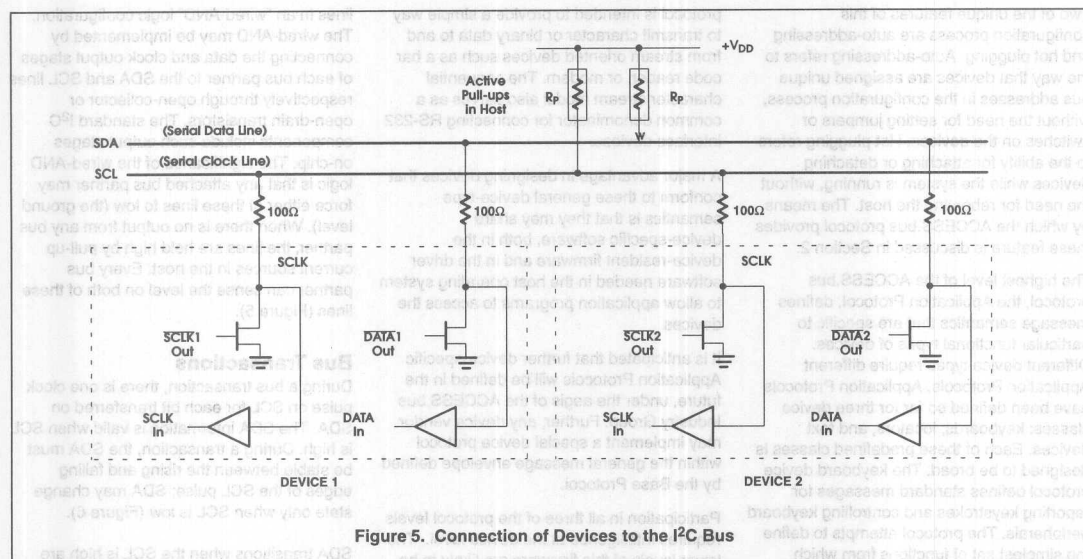


Figure 4. ACCESS.bus protocol hierarchy

## ACCESS.bus™ technical overview



## ACCESS.bus™ technical overview

### Synchronization

When a bus partner wishes to assert mastership of a free bus, it generates a start condition by pulling the SDA low. When SDA is low the new master begins the clock cycle, pulling the SCL low. All bus partners must be able to sense these events, and they respond by pulling all their SCL outputs low and beginning to count off their low periods. When each bus partner has reached the end of its low period, it lets its SCL output go high. Thus the SCL line will remain low for the duration of the longest low clock period among the bus partners.

When all the bus masters have reached the end of their low periods and let their SCL outputs go high, then the SCL goes high. All bus partners must be able to sense this event, and they begin counting their high period. The first bus master to reach the end of its high period pulls the SCL low again. In this way, all the bus partners simultaneously communicating on the bus are synchronized by a clock pulse whose low is as long as the longest of the low periods and whose high is as long as the shortest of the high periods. This synchronization persists until the master relinquishes the bus by generating a stop condition. The cooperative synchronization is a mechanism by which devices with slower clocks can regulate the operating rate of the bus. However, this mechanism, called "clock stretching", is not the normal means of data stream flow control. The ACCESS.bus protocol provides another mechanism for this purpose. (See Section 3.3.)

### Byte Framing and Acknowledgement

During this synchronized exchange the master/transmitter puts data on the SDA, one bit for each clock pulse. Eight successive bits comprise a byte, the most significant bit going first. The ACCESS bus is a Big Endian System.

As the new master puts the first byte on the bus, all the other bus partners participate in the synchronization. The first byte of the transaction contains the address of the intended slave/receiver of the transaction. Each non-master can check the address bits as they appear and cease participating in the synchronization as soon as the address bit on SDA fails to match the corresponding bit of its own address. The address check may also be done at the end of the address transmission.

At the end of the first byte, the master/transmitter lets its data output go high for the next clock pulse and the slave/receiver whose address matches the

transmitted address, is obliged to acknowledge receipt of the byte by pulling the SDA low for this pulse. This 1-bit Ack continues after each byte of the bus transaction; the master lets its SDA output go high and the receiver must pull the SCL low. Failure of the receiver to acknowledge a byte is an exception condition, which requires the master to terminate the transaction.

### Addressing

The slave/receiver of the bus transaction is determined by the address contained in the first byte. I<sup>2</sup>C uses the seven high-order bits of the first byte for addressing, and bit 0 to indicate whether the master is transmitting or receiving data. In ACCESS.bus, the master is always the transmitter, so bit 0 of the first byte of a transaction is always 0. Of the 128 7-bit addresses ACCESS.bus uses 15 addresses for general microcontrollers.

The host computer address is always 50h. In the configuration process to be described below, each peripheral device is assigned a unique address from the set of even numbers. 6Eh is used as a default address for devices before they have been assigned a unique address. A total of 125 addresses are available for devices on the bus.

### Arbitration

What happens when two devices simultaneously assert mastership? While putting data on the SDA, each transmitting master is, of course, independently sensing the state of SDA. Whenever a contending master detects that the state of the SDA is different from the data value it is putting out during a clock high, the contending master backs off, and waits for the stop condition before trying again. Thus, two contending masters will both put data on the bus as long as they are putting out the same data. The first bit where they differ will cause the contender that put out a 1 to back off.

Thus, under normal expected operation contending masters trying to send to different bus addresses will resolve the contention by the end of the first byte of the bus transaction. In the ACCESS.bus Base Protocol, the second byte of a transaction is the address of the transmitting master. Thus, as long as bus addresses are unique, the mastership of the bus will be resolved by the end of the second byte of the transaction. However, if two devices have the same address and are trying to send identical messages to a common address, then they will both send the entire message in unison. This situation can happen only during the configuration process before devices have all been assigned unique addresses; it is discussed further in Section 2.9 below.

Note that this arbitration mechanism never causes lost data or wasted transmissions, since the addresses of the receiver and transmitter are necessary overhead, in any case, for any sensible bus protocol. Note that bus priorities during arbitration are fixed by the device addresses, first by the address of the receiver, and then, for messages addressed to a common receiver, by the address of the transmitter. Lower addresses have priority over higher addresses. Lockouts of devices with high addresses are prevented by a rule of the Base Protocol that requires partners to wait a minimum time after relinquishing mastership before asserting it again.

### Message Format

An ACCESS.bus message comprises one I<sup>2</sup>C bus transaction. It consists of a string of bytes sent by a master/transmitter, each byte acknowledged by a one bit SCL-low Ack from the slave/receiver. The entire transaction is delimited by start and stop conditions generated by the master.

The first byte in the message is the receiver's unique address, as described above in Section 2.5. The second byte contains the transmitter's unique address.

The third byte of an ACCESS.bus message comprises two fields. Bits 2-7 provide a byte count for the body of the message. Thus, a message body can have 0 to 127 bytes. The message body is followed by a checksum byte, for error control. The checksum is the bitwise XOR of all the preceding bytes of the message.

The high order bit of the third byte is a Protocol Flag (P) to distinguish between data stream messages (P=0) and control/status messages (P=1). The data stream messages carry the application information being exchanged between the device and the host. The control/status messages are used to manage the ACCESS.bus protocol (Figure 8).

### Control/Status Messages

The ACCESS.bus Base Protocol defines a number of control/status messages that pertain to the Interface Parts of devices. These control/status messages are used for the configuration process, in which devices are assigned unique bus addresses and connected with the appropriate drivers in the host. The configuration process is described in Section 2.9 to 2.11. In a control/status message, the message type is indicated by an operation code contained in the first byte of the message body. The various Interface Part control/status messages are shown in Table 1.



## ACCESS.bus™ technical overview

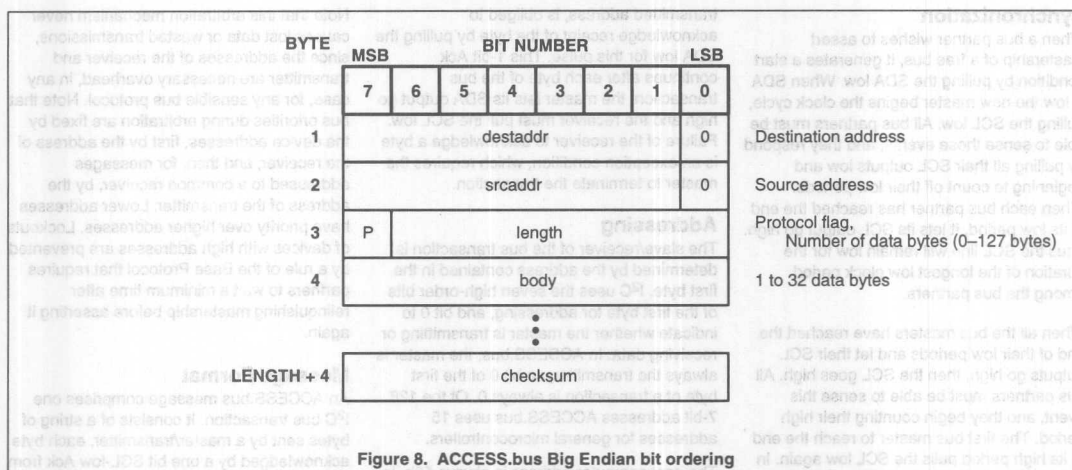


Figure 8. ACCESS.bus Big Endian bit ordering

Table 1. Interface Part Control/Status Messages

## Computer-to-Device Messages Purpose

Reset()	Force device to power-up state and default I <sup>2</sup> C address.
Identification Request()	Ask device for its "identification string."
Assign Address(ID string, new addr)	Ask device for its "identification string" to change its address to "new address."
Capabilities Request(offset)	Ask device to send the fragment of its capabilities information that starts at "offset."
Attention(status)	Inform computer that a device has finished its power-up/reset test and needs to be configured; "status" shall be the test result.
Identification Reply(ID string)	Reply to Identification Request with device's unique "identification string."
Capabilities Reply(offset, data frag)	Reply to Capabilities Request with "data fragment," a fragment of the device's capabilities string; the computer uses "offset" to reassemble fragments.
Interface Error()	Invalid checksum or premature end of message detected.

In addition, the Application Protocols define further control/status messages that are specific to particular device types. Some of these are discussed in Section 3 on the predefined device types.

## AppHardSiga

Provision for a subdevice to generate an interrupt of the host system.

## AppTest

A message from the host to a peripheral device commanding it to test the Application Part subdevice.

## AppTestReply

A message by which a device replies to an AppTest command.

## Configuration

The ACCESS.bus features auto-addressing and hot-plugging. These features are supported by the ACCESS.bus configuration process, which uses the seven types of control/status messages. Configuration consists of assigning unique bus addresses to the attached devices and connecting them with the appropriate drivers to provide host-resident application programs with access to the devices.

Configuration occurs when the device is powered-up or when it receives a Reset message. When the system is powered up, so are devices attached to the ACCESS.bus. Otherwise, devices are powered-up when they are hot-plugged into the bus. A device

may have a power source other than the +5V power line of the ACCESS.bus, but it must also be able to sense this line voltage and enter the power-up state when attached to the ACCESS.bus power source. When the system completes its boot up, the host sends a Reset message to all legal device addresses to put all devices in the power-up state.

Usually, when a device is powered up, it performs its specific self-testing. At the conclusion of self-test the device must assume the default address 6Eh and send an Attention message to announce its presence to the host. This message contains a single status byte to inform the host of the results of the power-up self-testing; zero indicates normal results and non-zero values indicate exception conditions that are specific to the device type.

On receiving an Attention message, the host sends an IDRequest message to the default address. Each device at this address replies with an IDReply message containing a unique 28-byte ID string described in the next section. The host is then able to assign unique ACCESS.bus addresses to each of the devices at the default address, by sending AssignAddress messages to the default address. Each AssignAddress message contains in its body the assigned address and the unique ID string of the corresponding device.



## ACCESS.bus™ technical overview

### Device Identifiers

During configuration, before address assignment, each device can be identified by a 28-byte unique ID string, which may be partly or entirely encoded in the device ROM. The first 24 bytes of the ID string are understood as ASCII-encoded information characterizing the device type:

- protocol revision 1 byte ("A")
- module revision 7 bytes (e.g., "V1.0 ")
- vendor name 8 bytes (e.g., "DEC ")
- module name 8 bytes (e.g., "LK501 ")
- device number 32-bit signed integer

The first 24 bytes characterize the device's firmware and are encoded in the device ROM.

The remaining 4 bytes of the device ID string are understood as a 32-bit two's complement integer that uniquely identifies the device among devices of the same type. This integer may be provided as a unique serial number contained in the device ROM. Or, in the absence of such a serial number, interactive devices may use a random or arbitrarily determined number for this part of the ID string. As an aid to the host software, the Base Protocol specifies that, in the IDReply message, unique serial numbers be sent as positive integers and randomly generated numbers be sent as negative integers, in the two's complement sense.

In the random number case, it is possible that different devices of the same type may come up with the same 32-bit discriminator. In this case, the different devices will be assigned the same bus address, an undesirable situation. The ACCESS.bus specification suggests a guideline to help avoid identical random identifiers: use the number of cycles of the device's own clock between power up and the time the IDRequest message is received. The natural dispersion of the frequencies of these oscillators is likely to provide unique numbers.

The Base Protocol includes a provision to ensure against the unlikely circumstance that different interactive devices of the same type and without unique serial numbers will generate the same random number. Namely, each such device must send a Reset message to its own assigned address. This self-addressed Reset is sent only once between power-ups or external Resets, just before the device sends the first message instigated by a user action. Of course, the transmitting device will ignore the self-addressed Reset, but other devices

possibly at the same address will be reset and will go through configuration again. The Base Protocol specifies that a device using a random number in its ID string shall change that random number after receiving a Reset message. In this way, all interactive devices are guaranteed assignment of unique addresses before sending their first data-carrying messages.

If several non-interactive devices of the same type are to be attached to a single ACCESS.bus, then they must have hardwired unique serial numbers.

During normal operation, the host periodically checks the configuration by sending IDRequest messages to inactive devices.

The host also sends IDRequest messages to all assigned addresses whenever it receives an Attention message from a device seeking configuration. The purpose of these IDRequest messages is to verify the current state of the ACCESS.bus – what devices are still connected and which devices are no longer present.

### Device Capabilities Information

In order that a device be accessible to application programs running on the host, it must be connected to an appropriate software driver. Establishing this association is the last phase of configuration.

The appropriate driver will depend on the device type. There may be further parameters that characterize the device and which affect the choice of driver, or which at least must be furnished as arguments to the selected driver. Moreover, the application program may also need to be informed of these device parameters. The device Capabilities Information feature of the ACCESS.bus protocol allows a measure of device independence in the selection of drivers and provides for informing the host software of the device characteristics.

Device Capabilities Information is an explicit statement of a device's functional characteristics that are only implicit in the device type designation contained in the ID string, or that may even vary among individual devices of a given type. For example, the Capabilities Information about a keyboard might include the national alphabet used, or the Capabilities Information about a locator might include its resolution or units.

The Capabilities Information for each device is contained in a single human-readable ASCII-encoded text string stored on the device ROM. The ACCESS.bus Base Protocol defines a simple and compact grammar for building the capabilities string.

The semantics of the Capabilities Information is carried by keywords. The Base Protocol defines some keywords that can apply to all sorts of devices. Then each Application Protocol will define further keywords that are meaningful only for certain types of devices. To date, the ACCESS.bus Application Protocols define semantics for the Capabilities Information for generic keyboards, locators, and general text devices. The grammar allows for easy extension of the Capabilities Information specification.

An example of a simple mouse Capabilities string might be as follows:

```
(
  prot(locator)
  type(mouse)
  model(VSXXX)
  buttons(1(L)2(R)3(M)) dim(2) rel
  res(200 inch) range(-127 127)
  d0(dname(X))
  d1(dname(Y))
)
```

This would specify that the device uses the standard locator protocol; that it is a mouse, that it is model VSXXX; that it has three buttons designated left, middle and right corresponding with the respective bits in the keyswitch word; that it has two degrees of freedom, designated "X" and "Y", using inches as units with 200 counts per inch; that it reports relative values (displacement since last report) in the range -127 to 127 counts, and that its coordinate values correspond to X and Y.

After assigning a unique address to a device, the host sends it a CapRequest message to command it to send its Capabilities Information in a CapReply message. Of course, the device Capabilities string may well exceed the capacity of the message body of each CapRequest specifies where in the Capabilities Information string the fragment should start. The 'offset' value is repeated in the CapReply message, to be used as a check by the host in reassembling the Capabilities string. The offset is restricted to three values: send first (0), send again, and send next (offset from most recent string plus the number of bytes in the fragment).

The prot(), type(), and model() keywords must appear in that order and occur first in the Capabilities string. They must be within the first 128 bytes.

## ACCESS.bus™ technical overview

### APPLICATION DEVICE TYPES

The initial ACCESS.bus specification defines Application Protocols for three kinds of devices: keyboards, locators, and text devices. An important advantage of developing devices that conform to these defined protocols is the availability of pre-existing software to implement them, in particular, the drivers in the operating software of the host system through which application programs gain access to the devices.

### Keyboard Devices

A generic keyboard consists of an array of key stations assigned numbers between 8 and 255. When any key station transitions between open and closed, the entire list of key stations currently closed or depressed is transmitted to the host.

In addition to reporting key stations, the generic keyboard device can support simple feedback mechanisms such as keyclicks, bells, and light-emitting diodes. These mechanisms are controlled explicitly from the host so that minimal keyboard state modeling is required. The keyboard mapping table can also be stored in the keyboard itself as part of the capabilities string.

Each key is assigned a unique 8-bit number (8-255). The first 8 codes are reserved for other keyboard functions. On each key transition, up or down, the keyboard will report the complete state of the key array as a list of zero to ten key stations that are currently down.

Example: user enters the modified keystroke Alt-Shift-A

Transition	Report
Alt down	Alt
Shift down	Alt Shift
'A' down	Alt Shift A
'A' up	Alt Shift
Shift up	Alt
Alt up	<empty list>

This reporting scheme is functionally complete in that the host can detect every key transition and it provides the full state of the keyboard on each report. No special resynchronization reports are needed.

More detailed information can be found in the ACCESS.bus Keyboard Device Protocol Specification.

### Locator Devices

The *ACCESS.bus Locator Device Specification* provides for a device that has up to 15 degrees of freedom (with 16-bit precision) and up to 16 binary keys or buttons. Thus, in addition to such conventional pointer/locator devices as mouse, tablet, trackball. The ACCESS.bus locator protocol is suitable for valuator sets, such as dial boxes, and function key boxes with up to 16 function keys.

The locator capabilities information provides for specifying the number of switches and their designations (for example, "left", "right", "middle", etc.), whether the locator values are relative (like a mouse) or absolute (like a tablet or dial box), the resolution (counts per unit, and units), the dynamic range, and the names of the locator axes (for example, "x", "y", etc.).

The first 2-byte word of the event report message body contains a mask giving the state of the switches; the remaining words contain the value of each of the locator axes, either the absolute values or the change since the previous report in the case of relative devices. The locator report message is sent either on a regular sampling interval or on receipt of an AppPoll message from the host.

The locator-specific Application Protocol control/status messages are from the host to the device:

- AppPoll**  
Requests the device to report its state
- AppSamplingInterval**  
Sets the device sampling interval or instructs the device to report only when polled

More detailed information can be found in the ACCESS.bus Locator Device Protocol Specification.

### Text Devices

The text device protocol is intended to provide a simple way to transmit character or binary data to and from stream oriented devices such as a bar code reader, or character display. The sequential character stream model also serves as a common denominator for connecting RS-232 interfaced devices.

A generic text device transmits a stream of 8-bit bytes from a character set. Simple control messages are defined to support flow control and to select communication parameters that might be used to interface with a modem. The capabilities string

contains information that identifies the specific character set and communication parameters used.

More detailed information can be found in the ACCESS.bus Text Device Protocol Specification.

### TIMING RULES

To ensure good interactive response and to ensure that all devices will have access to the bus, the ACCESS.bus specifies rules on transaction timing. Further, specific timeouts are needed to avoid hanging up the interconnect when devices fail or are removed.

### Transaction Timing Rules

The ACCESS.bus is designed primarily for interactive devices. A basic objective of the definition of the ACCESS.bus specification is that every interactive device should be able to update the host on its state at least once in every display video frame time. To help meet this criterion, the Base Protocol imposes some rules on the timing of a device's interaction with the bus.

### Response Timeouts

In order that a dead or unplugged device will not hang up the system indefinitely, there must be time limits for responding to commands that require a response. The ACCESS.bus protocol specifies that devices shall complete the Reset command within 250ms. Further, a device shall respond to any command requiring a response within 40ms, or, in the case of commands that can be answered by several devices, within 40ms after the last device to respond.

### SOFTWARE ARCHITECTURE AND DEVELOPMENT

An ACCESS.bus peripheral requires software at both ends of the bus transaction for managing all levels of the peripheral interaction: the I<sup>2</sup>C interface, the ACCESS.bus Base Protocol, and the ACCESS.bus Application Protocol. Further, the peripheral device requires software to support communication between the device microcontroller and the application-specific I/O transducer circuitry. Finally, the host system operating software must provide interfaces by which application programs can access both the ACCESS.bus devices and the ACCESS.bus itself. An important advantage of the ACCESS.bus approach is that the lower levels of the interaction are common to diverse device types, so they can be supported by the same or similar software modules.

## ACCESS.bus™ technical overview

### Device Firmware Development

The microcontroller in the device provides the intelligence for managing the device's participation in all the levels of the ACCESS.bus protocol. Use of the components with hardware I<sup>2</sup>C interface functionality described in "MICROCONTROLLERS", can simplify the development of the lowest level of this software. Moreover, because they concern only the bus communication methods that are common to all sorts of peripheral devices, the I<sup>2</sup>C interface and the ACCESS.bus Base Protocol may well be implemented by reusing software previously developed for some of these components. And devices conforming to the semantics of the predefined standard Application Protocols may also benefit from the availability of some off-the-shelf code at the top level.

Of course, each device vendor will have to develop substantial firmware specific to his or her device. Philips and several third party vendors offer a range of tools to support firmware development for the standard components of the 80C51 family. These tools include cross assemblers and cross compilers for C and PL/M, in-circuit emulators with symbolic debugging and real-time trace support, and EPROM programming equipment. Generally, these software and hardware tools are for use with PC-compatibles as the development platforms.

### Host Software Architecture

Vendors of host systems supporting ACCESS.bus will have to supply drivers and other kernel modules to provide access to the ACCESS.bus port, both for application program clients and for other system software, such as the interactive I/O handlers of the window system.

### DEVELOPMENT SUPPORT

Both Digital and Philips Semiconductors offer technical support and assistance to developers of ACCESS.bus devices and host systems.

### ACCESS.bus Industry Group

The ACCESS.bus Industry Group (ABIG) is an association of members interested in

promoting ACCESS.bus as an industry standard for the desktop connectivity of computer peripherals. As an association, ABIG is intended to maintain the ACCESS.bus specification as a simple, easy-to-implement, stable technology in the spirit of its design and contribute to the technical longevity of the ACCESS.bus architecture.

ABIG is an open industry group and anyone who has an interest in ACCESS.bus can be an ABIG member. An ABIG member is defined as a company including its divisions and subsidiaries, an organization, or a public or private institution.

There are two basic types of membership; General and Voting. General membership is open to everyone and Voting Membership is restricted to only those members who are actively developing a device or platform that incorporates the ACCESS.bus technology. ABIG is governed by an elected Steering committee of seven voting members.

ABIG Founding Members are those voting members who joined to form ABIG. They have the same voting privileges. The ABIG Founding Members are:

AMP, Inc.  
Ceibo, Ltd.  
Computer Access Technology Corp.  
Digital Equipment Corp.  
Discrete Time Systems Corp.  
Honeywell Keyboard Division  
Input Technologies, Inc.  
ITAC Systems, Inc.  
Kensington Microware Limited  
Lexmark International, Inc.  
Logitech, Inc.  
Micro Computer Control Corp.  
Molex Inc.  
Mouse Systems Corp.  
New Idea Electronic Co., Ltd.  
Nexus Applied Research, Inc.  
Penny & Giles Computer Products Ltd.  
Philips Semiconductors  
Robert Clemens Research & Development  
Summagraphics Corp.  
Sun International, Inc.  
Welch Allyn, Inc.

For more details on ABIG, please contact ABIG directly at:

ACCESS.bus Industry Group  
370 Altair Way, suite 215  
Sunnyvale, California 94086  
Telephone: 408-991-3517  
FAX: 408-991-3773

### Philips Semiconductors Support

An ACCESS.bus specification is available from Philips Semiconductors. This kit includes the complete specifications for the ACCESS.bus Base Protocol and pre-defined Application Protocols, as well as the Philips Data Handbook containing the detailed specification of the I<sup>2</sup>C bus, characteristics of the available integrated circuits which support it, application notes, and sample firmware code. The Data Handbooks also contain listing of development systems and third-party products supporting microcontroller firmware development, mentioned in the section entitled "Device Firmware Development".

Beside the 80C51-family microcontrollers, Philips and other manufacturers offer over 100 different components with built-in I<sup>2</sup>C support: memories, display controllers, data converters, clock/calendars, voice synthesizers, video processors, and others.

For technical questions on I<sup>2</sup>C call your local Philips Semiconductors sales office, or contact the Philips' Headquarters Application Group at [1]408-991-3518.



**Purchase of Philips' I<sup>2</sup>C components conveys a license under the Philips' I<sup>2</sup>C patent to use the components in the I<sup>2</sup>C-system provided the system conforms to the I<sup>2</sup>C specifications defined by Philips.**



## **ACCESS.bus Development Kit**

## **A.b-DEV-KIT**

ACCESS.bus is an open industry standard providing a simple and uniform way to connect up to 125 devices to a single port on a computer. ACCESS.bus features 100,000 bits per second data rate, hardware arbitration, dynamic reconfiguration, a comprehensive capabilities grammar to support generic software device drivers, and off-the-shelf, low-cost I2C Microcontroller technology.

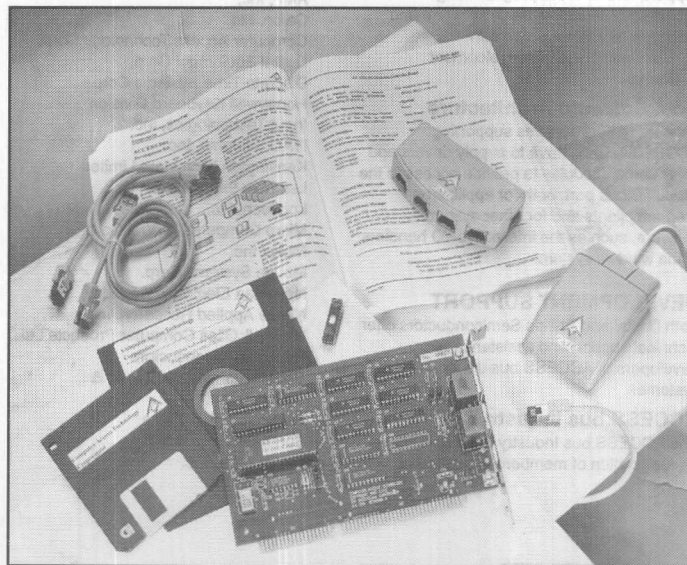
The A.b-DEV-KIT is an ACCESS.bus product development support package. It includes CATC's PC/AT A.b-125I ACCESS.bus controller board, an ACCESS.bus mouse, expansion box and cables and an 87C751 Microcontroller. The kit also includes a comprehensive software package, a user's manual and Hotline telephone support.

The software package includes on-board microcode, an ACCESS.bus Manager that runs as a TSR under DOS and an ACCESS.bus Monitor and Control program. In addition the A.b-DEV-KIT software includes source code of ACCESS.bus generic software drivers for the host and ACCESS.bus devices software modules.

### **Features**

- Complies fully with the ACCESS.bus standard
- ACCESS.bus hardware package including -
  - The A.b-125I PC/AT ACCESS.bus Controller
  - An A.b Mouse
  - An A.b expansion box
  - A.b cables (2 ft and 4 ft)
  - A Philips 87C751 Microcontroller
- A comprehensive software package including -
  - On-board ACCESS.bus Main Controller (MC) microcode
  - An ACCESS.bus Manager that runs as a TSR under DOS
  - An ACCESS.bus Monitor and Control program
  - Source code of an ACCESS.bus host generic software driver
  - Source code of a generic ACCESS.bus application layer software
  - Source code of ACCESS.bus devices software modules

- A comprehensive A.b user's manual
- Hotline telephone support



**The A.b-DEV-KIT includes ACCESS.bus accessories  
and comprehensive software**





## A.b-DEV-KIT

### A.b-125I ACCESS.bus Controller Board

#### ACCESS.bus Interface

Controls a standard ACCESS.bus network. Provides two industry standard ACCESS.bus connectors, supplying 5V @ 0.75 A.

User selectable interrupt -

IRQ10, IRQ11 or IRQ12

#### Physical

Power: +5V DC, 10 W max.

Temp. Range: +0 to +50 degree C

Board Size: 4.2" H x 6.5" W.

#### ACCESS.bus Network Size

Supports up to 125 ACCESS.bus devices. Physical distance up to 25 feet. With an external ACCESS.bus Buffer (optional) up to 250 feet.

#### System Interface

IBM PC/AT and compatibles. Uses the PC/AT 16-bit programmable input / output mechanism:

#### Warranty

90-day. Return to factory for repair or replacement at manufacturers option

User selectable I/O addresses -

0x250 to 0x25F

0x260 to 0x26F

0x350 to 0x35F

### A.b-DEV-KIT Software

#### On-board MC microcode

A real-time package that controls the operation of physical ACCESS.bus devices.

#### Source Code License

Source code of a generic ACCESS.bus software driver for the host. Source code of a generic ACCESS.bus application layer software. Source code of ACCESS.bus physical devices software modules.

#### ACCESS.bus Manager

Runs as a TSR under DOS, communicates with the MC microcode and with the various device drivers. It routes control and application messages between the physical devices and their respective software drivers.

#### Diagnostics

A comprehensive self test is performed on the board on power up. Diagnostics are run under control of the ACCESS.bus Monitor.

#### ACCESS.bus Monitor

A user-friendly, menu-driven program, displays user-selected ACCESS.bus messages and allows the user to control specific devices.

Product specifications are subject to change without notice

### Computer Access Technology Corporation

949 Hillsboro Avenue, Sunnyvale, CA 94087

Tel: (408) 732 8910 Fax: (408) 730 1675





## ACCESS.bus PC/AT Controller Board

A.b-125I

### Description

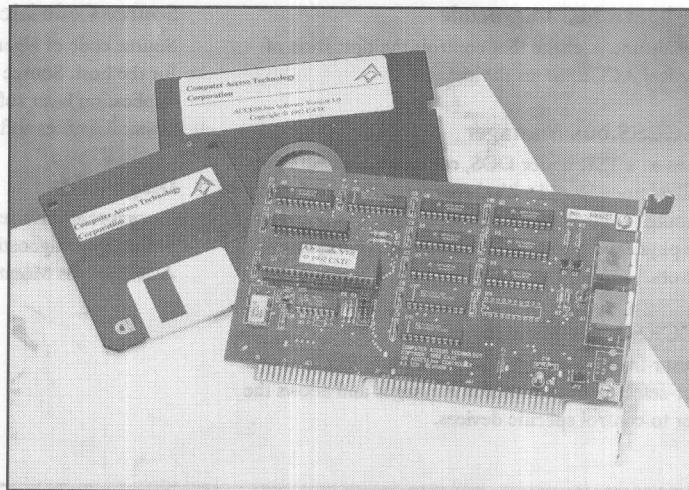
ACCESS.bus is an open industry standard providing a simple and uniform way to connect up to 125 devices to a single computer port. ACCESS.bus features data rate of 100,000 bits per second, hardware arbitration, dynamic reconfiguration, a comprehensive capabilities grammar to support generic software device drivers, and off-the-shelf, low-cost I2C microcontroller technology.

The A.b-125I is a PC/AT adapter board that serves as an ACCESS.bus master. It allows the connection of multiple devices to a single port on the PC. The board can be used in Desktop connectivity as well as in Control and Instrumentation applications. The board is based on a Philips 8xC654 microcontroller with I2C interface.

The A.b-125I is offered with a comprehensive software package including on-board microcode and an ACCESS.bus Manager that runs as a TSR under DOS.

### Features

- A highly integrated, half board design, uses a single 16-bit AT/ISA slot
- Full compliance with the ACCESS.bus standard
- On-board 8K bytes SRAM buffer memory
- A comprehensive software package including -
  - An on-board ACCESS.bus Main Controller (MC) microcode
  - An ACCESS.bus Manager that runs as a TSR under DOS



The A.b-125 allows the connection of multiple devices  
to a single port on the PC

**ACCESS.bus Interface**

Controls a standard ACCESS.bus network. Provides two industry standard ACCESS.bus connectors, supplying 5V @ 0.75 A.

**ACCESS.bus Network Size**

Supports up to 125 ACCESS.bus devices. Physical distance up to 25 feet. With an external ACCESS.bus Buffer (optional) up to 250 feet.

**Buffer Memory**

8K x 8 bits (8K bytes) static RAM,

**System Interface**

IBM PC/AT and compatibles. Uses the PC/AT 16-bit programmable input / output mechanism:

User selectable I/O addresses -

0x250 to 0x25F

0x260 to 0x26F

0x350 to 0x35F

User selectable interrupt -

IRQ10, IRQ11 or IRQ12

**Software**

A comprehensive software package is provided with the board. The software includes the on-board ACCESS.bus Main Controller (MC) microcode and the ACCESS.bus Manager that runs as a TSR under DOS.

CATC has additional ACCESS.bus software available including a Windows 3.1 version of the ACCESS.bus Manager, software device drivers, bus monitoring and control program and development tools. Call CATC for additional information.

**Diagnostics**

A comprehensive self test is performed on the board on power up.

**Physical**

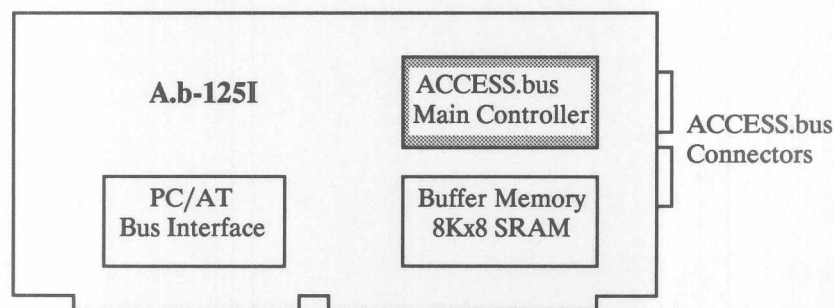
Power: +5V DC, 10 W max.

Temp. Range: +15 to +50 degree C

Board Size: 4.2" H x 6.5" W.

**Warranty**

90-day. Return to factory for repair or replacement at manufacturers option



Product specifications are subject to change without notice

**Computer Access Technology Corporation**

949 Hillsboro Avenue, Sunnyvale, CA 94087

Tel: (408) 732 8910 Fax: (408) 730 1675



## Section 6

### Control Area Network (CAN) Bus

80C51-Based  
8-Bit Microcontrollers

#### CONTENTS

Control Area Network (CAN) overview .....	1185
<b>82C150</b> CAN serial linked I/O device with digital and analog port functions .....	1186
<b>82C200</b> Stand-alone CAN-controller .....	1207
<b>82C250</b> CAN controller interface .....	1243





## Control Area Network (CAN) overview

### CAN OVERVIEW

The Control Area Network (CAN) is a multiplexed wiring protocol developed by Bosch for use in automotive applications. As products supporting CAN have been made available by Philips Semiconductors and other semiconductor manufacturers, the protocol has become used in other industries including: industrial automation, machine tools, medical equipment, and building environmental control, to name a few. The CAN protocol is attractive for use in a wide

range of applications because it has powerful error detection capabilities and features differential drive, and can be used comfortably in critical high noise environments. CAN is also very flexible in terms of the transmission media and the connection scheme, and is generally easy to adapt to most applications.

Philips offers a wide range of parts that support the CAN protocol, including

stand-alone parts as well as microcontrollers with integrated CAN interfaces. Datasheets for the 82C200 (Stand-alone CAN controller), 82C150 (CAN serial Linked I/O device), and 82C250 (CAN transceiver) are included in this section of this databook. Datasheets for the microcontrollers that have an integrated CAN interface (8XC592 and 8XC598) are included with Philips' 80C51 family products in Section 3 of this databook.

- Up to 10 digital inputs
- Automatic transmission of a CAN message on a change on inputs individually selectable
- Up to 10 3-state outputs
- Up to two quasi-serial outputs with 10-bit accuracy
- 10-bit analog-to-digital converter with up to six multiplexed analog input channels
- Two general purpose comparators
- Bit rate from 50 kbaud up to 125 kbaud using internal oscillator (extension by external clock possible)
- Automatic bit rate detection and calibration
- Up to 16 82C150 nodes for one CAN-bus system
- Four identifier bits programmable
- Automatic recovery from bus yielding failures supported
- SLO functions controlled by a single intelligent node ("host")
- Sleep-mode with wake-up via CAN-bus
- Differential CAN-bus input comparator and CAN-bus output driver
- 5V ±4% supply voltage range
- Operating temperature range from -40 to +125 °C

- Applications:
- body electronics and instrumentation in automotive applications
  - sensor/actuator interface in automotive and general industrial applications
  - extension of I/O capabilities of microcontrollers

### 3. ORDERING INFORMATION

PART ORDER NUMBER PART MARKING	PHILIPS NORTH AMERICA PART ORDER NUMBER	PACKAGE			TEMPERATURE RANGE (°C)
		PINS	PIN POSITION	MATERIAL	
82C150 AH1	82C150 AH1D	28	8008	plastic	-40 to +125
82C150 AH2	82C150 AH2D	28	8008	plastic	-40 to +125

NOTE:  
1. Parts ordered by the Philips North America part number will be marked with the Philips part marking.

CAN serial linked I/O device  
with digital and analog port functions

82C150

1. FEATURES

- Single-chip I/O device with CAN protocol controller
- Meets CAN protocol specification version 2.0 A and B (passive) with restricted bit timing
- Fully integrated clock oscillator (no crystal required)
- 16 configurable digital or analog I/O port pins
- Each of the port pins individually configurable via CAN-bus: port direction, port mode and event capture facilities for inputs (event driven or polling)
- Up to 16 digital inputs: automatic transmission of a CAN message on a change on inputs individually selectable
- Up to 16 3-state outputs
- Up to two quasi-analog outputs with 10-bit accuracy
- 10-bit analog-to-digital converter with up to six multiplexed analog input channels
- Two general purpose comparators
- Bit rate from 20 kbit/s up to 125 kbit/s using internal oscillator (extension by external clock possible)
- Automatic bit rate detection and calibration
- Up to 16 P82C150 nodes for one CAN-bus system
- Four identifier bits programmable
- Automatic recovery from bus wiring failures supported
- SLIO functions controlled by a single intelligent node ("host")
- Sleep-mode with wake-up via CAN-bus
- Differential CAN-bus input comparator and CAN-bus output driver
- 5V  $\pm$ 4 % supply voltage range
- Operating temperature range from -40 to +125 °C

2. GENERAL DESCRIPTION

The P82C150 is a single-chip 16-bit I/O device including a controller area network (CAN) protocol controller with automatic bit rate detection and calibration. It features 16 configurable I/O port pins with programmable direction, digital and analog modes.

The P82C150 provides a configurable event capture facility supporting automatic transmission caused by a change on the port input pins.

The clock oscillator requires no external components, thus, the cost of the CAN link is reduced significantly.

The P82C150 is a very cost-effective way to increase the I/O capability of a microcontroller as well as to reduce the amount and complexity of wiring. Advanced safety is provided by the CAN protocol.

Applications:

- body electronics and instrumentation in automotive applications
- sensor/actuator interface in automotive and general industrial applications
- extension of I/O capabilities of microcontrollers.

3. ORDERING INFORMATION

PHILIPS PART ORDER NUMBER PART MARKING	PHILIPS NORTH AMERICA PART ORDER NUMBER <sup>1</sup>	PACKAGE				TEMPERATURE RANGE (°C)
		PINS	PIN POSITION	MATERIAL	CODE	
P82C150 AHT	P82C150 AHTD	28	SO28	plastic	SOT136A	-40 to +125

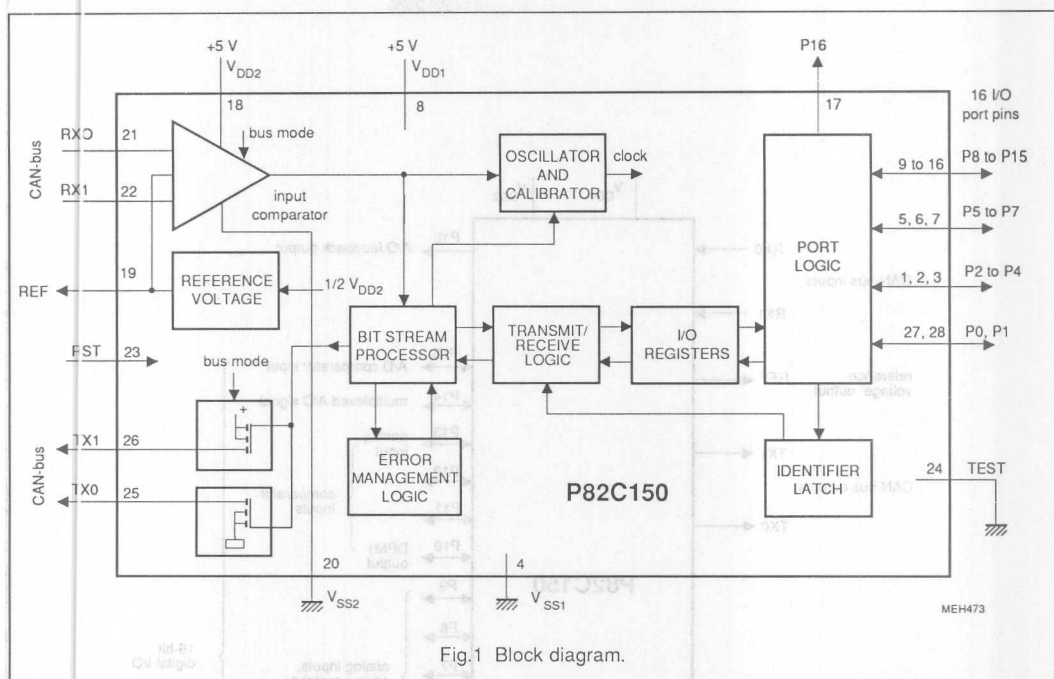
NOTE:

1. Parts ordered by the Philips North America part number will be marked with the Philips part marking.

# CAN serial linked I/O device with digital and analog port functions

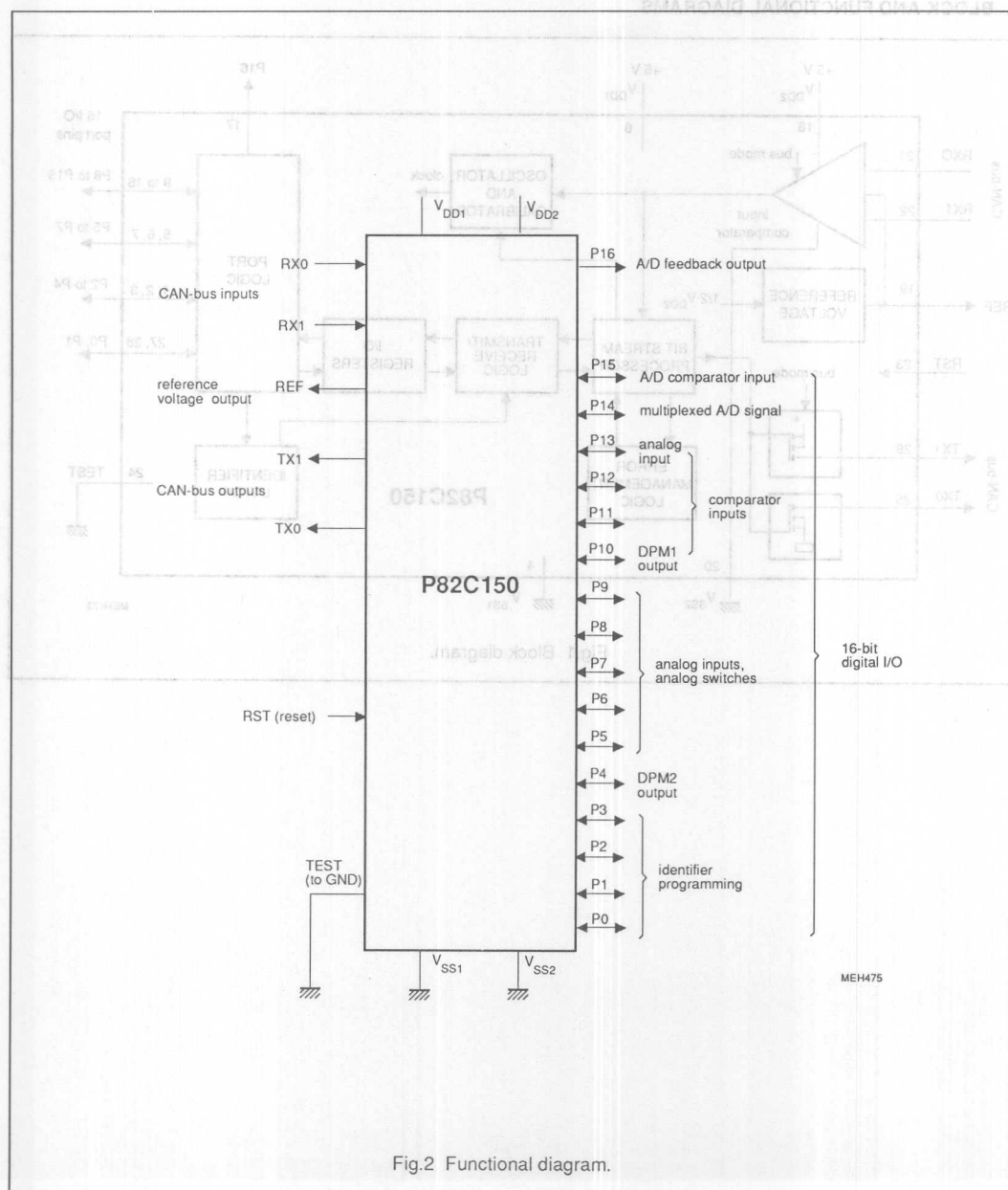
82C150

## 4. BLOCK AND FUNCTIONAL DIAGRAMS



# CAN serial linked I/O device with digital and analog port functions

82C150



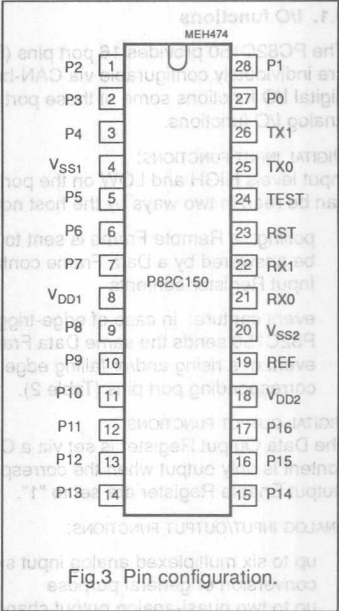
CAN serial linked I/O device  
with digital and analog port functions

82C150

5 PINNING

SYMBOL	PIN	DESCRIPTION
P2	1	I/O ports P2 to P3; Identifier programming input
P3	2	
P4	3	I/O port 4; DPM2 output
V <sub>SS1</sub>	4	logic ground (0 V; logic circuits and CAN-bus driver)
P5	5	I/O ports P5 to P6; analog input
P6	6	
P7	7	I/O port 7; analog input or A/D comparator 1 output
V <sub>DD1</sub>	8	+5 V supply voltage (logic circuits and CAN-bus driver)
P8	9	I/O port 8; analog input or comparator 3 output
P9	10	I/O port 9; analog input or comparator 2 output
P10	11	I/O port 10; comparator 3 inverting input or DPM1 output
P11	12	I/O port 11; comparator 3 non-inverting input
P12	13	I/O port 12; comparator 2 inverting input
P13	14	I/O port 13; comparator 2 non-inverting input
P14	15	I/O port 14; multiplexed analog signal
P15	16	I/O port 15; A/D comparator input
P16	17	feedback output of A/D converter
V <sub>DD2</sub>	18	+5 V supply voltage (CAN input, oscillator, reference)
REF	19	reference voltage output (1/2 V <sub>DD2</sub> )
V <sub>SS2</sub>	20	analog ground (0 V; CAN input, oscillator, reference)
RX0	21	CAN-bus input
RX1	22	
RST	23	external reset input (active-HIGH)
TEST	24	test input; connected to ground
TX0	25	open-drain CAN-bus output: dominant = LOW; recessive = floating
TX1	26	open-drain CAN-bus output: dominant = HIGH; recessive or at bus mode 2 floating
P0	27	I/O ports P0 to P1; Identifier programming input
P1	28	

PIN CONFIGURATION





# CAN serial linked I/O device with digital and analog port functions

82C150

## 6. FUNCTIONAL DESCRIPTION

### 6.1. I/O functions

The PC82C150 provides 16 port pins (P15 to P0) which are individually configurable via CAN-bus. Besides the digital I/O functions some of these port pins provide analog I/O functions.

#### DIGITAL INPUT FUNCTIONS:

Input levels HIGH and LOW on the port pins (P15 to P0) can be read in two ways by the host node:

- polling: a Remote Frame is sent to the P82C150 to be answered by a Data Frame containing the Data Input Register contents
- event capture: in case of edge-triggered mode, the P82C150 sends the same Data Frame caused by the event of a rising and/or falling edge on the corresponding port pins (Table 2).

#### DIGITAL OUTPUT FUNCTIONS:

The Data Output Register is set via a CAN message. Its content is only output when the corresponding bits of the Output Enable Register are set to "1".

#### ANALOG INPUT/OUTPUT FUNCTIONS:

- up to six multiplexed analog input signals for A/D conversion or general purpose
- up to two quasi-analog output channels (DPM)
- two input comparators, for example for window comparator applications
- a separate A/D input comparator with feedback output (10-bit  $\pm 2$  LSB A/D conversion)

A/D converted digital results are obtained by reading the A/D register. Analog functions of each port pin are

individually controlled by the Analog Configuration Register. Writing the I/O registers is done serially via CAN-bus by Data Frames. The first data byte contains the register address, and the second and third data bytes represent the register contents. If a read-only register is addressed, the contents of the second and third data bytes are ignored. It is recommended to set unused port pins defined HIGH (100 k $\Omega$  resistor to V<sub>DD</sub>).

#### 6.1.1. Data Input Register (address 0; read only)

This register contains the states of port pins P15 to P0 which are transmitted on request, or automatically by change of one of the input levels, provided that the respective input is configured to event capture mode (Table 2). When an edge is detected, the port state is loaded after sending the Control Field of the triggered message. Thereby a delay for input settling is provided. Additionally, the register content is sent automatically after a reset or bus mode change, once the bit time has been calibrated.

#### 6.1.2. Positive Edge Register (address 1; write only)

This register contains configuration information per port pin for the event capture facility. The corresponding PE-bit (Table 2) has to be set to "1" to enable capturing of the rising edge.

#### 6.1.3. Negative Edge Register (address 2; write only)

This register contains configuration information per port pin for the event capture facility. The corresponding NE-bit (Table 2) has to be set to "1" to enable capturing of the falling edge.

The combination of PE and NE functions is possible.

Table 1 I/O Register map

REGISTER	ADDRESS	MSB								LSB							
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data input	0	DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
Positive edge	1	PE15	PE14	PE13	PE12	PE11	PE10	PE9	PE8	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
Negative edge	2	NE15	NE14	NE13	NE12	NE11	NE10	NE9	NE8	NE7	NE6	NE5	NE4	NE3	NE2	NE1	NE0
Data output	3	DO15	DO14	DO13	DO12	DO11	DO10	DO9	DO8	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
Output enable	4	OE15	OE14	OE13	OE12	OE11	OE10	OE9	OE8	OE7	OE6	OE5	OE4	OE3	OE2	OE1	OE0
Analog configuration	5	ADC	OC3	OC2	OC1	0	M3	M2	M1	SW3	SW2	SW1	0	0	0	0	0
DPM1	6	DP9	DP8	DP7	DP6	DP5	DP4	DP3	DP2	DP1	DP0	0	0	0	0	0	0
DPM2	7	DQ9	DQ8	DQ7	DQ6	DQ5	DQ4	DQ3	DQ2	DQ1	DQ0	0	0	0	0	0	0
A/D	8	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	0	0	0	0	0	0

# CAN serial linked I/O device with digital and analog port functions

82C150

**Table 2** Programming of the I/O registers to event capture on edge or to digital output function (X = don't care)

REGISTER CONTENTS OF PARTICULAR PORT PIN		DIGITAL INPUT FUNCTION				DIGITAL OUTPUT FUNCTION
		POLLING	EVENT CAPTURE ON EDGE		RISING AND FALLING	
	bits		RISING	FALLING		
Positive edge	PE	X	1	0	1	X
Negative edge	NE	X	0	1	1	X
Output enable	OE	X	X	X	X	1

**6.1.4. Data Output Register (address 3; write only)**

This register contains the output data for the port pins. The output drivers are bitwise enabled by OE.

**6.1.5. Output Enable Register (address 4; write only)**

This register controls the output drivers of the port pins. The corresponding Output Enable Register bit has to be set to "1" to enable an output driver. If set to "0", the corresponding output driver is disabled (floating; Fig. 4).

**6.1.6. Analog Configuration Register (address 5; read/write)**

This register contains the bits ADC, OC3 to OC1, M3 to M1 and SW3 to SW1 (Fig. 5).

ADC BIT (A/D conversion start bit; write-only bit):

The P82C150 starts an A/D conversion cycle at ADC = 1, ended with the transmission of a message containing the result. After that, the ADC bit is reset automatically.

OC3 to OC1 BITS (comparator output data; read-only bits):

The P82C150 sends back the logical output value of these comparators after having received a Data Frame (6.2.3.) addressing the Analog Configuration Register. Monitoring is also possible by using the event capture facility.

M3 to M1 BITS (multiplexer control bits; write-only bits):

The logical value of the comparators is monitored on port pins P8, P9 and P7 (Fig. 5) by setting M3 to M1 to "1", provided that these pins are configured as outputs (OE = 1).

SW3 to SW1 (analog switch control bits; write-only bits):

One of the analog switches S1 to S6 can be closed by setting the switch bits to the corresponding value (Fig. 5):

SW3	SW2	SW1	
0	0	0	no switch closed (S0)
0	0	1	S1 closed
0	1	0	S2 closed
0	1	1	S3 closed
1	0	0	S4 closed
1	0	1	S5 closed
1	1	0	S6 closed
1	1	1	reserved

Evidently if P14 is driven, it may not be connected to any other driven pin via the internal analog switches (avoid short-circuit!).

**6.1.7. DPM1 Register (address 6; write only)**

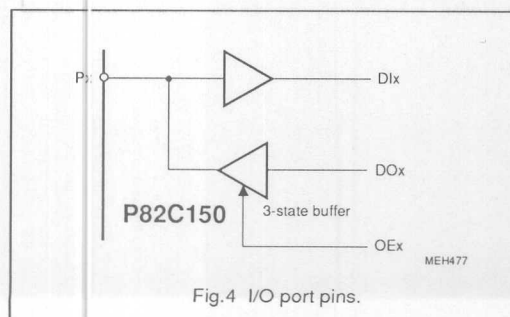
A quasi-analog output signal on port pin P10 is generated by distributed pulse modulation (DPM; Fig. 8) if the Output Enable bit is set (OE10 = 1). The DPM1 output signal is inverted by setting DO10 = 1. The number of output pulses during a DPM period is given by the DPM1 Register value. These pulses have  $4 \times t_{CLK}$  length and are distributed over the DPM period (Fig. 8). An analog voltage is provided after smoothing the output signal by an external RC combination.

**6.1.8. DPM2 Register (address 7; write only)**

The function of the DPM2 output (P4) and the DPM2 Register correspond to the definition of DPM1.

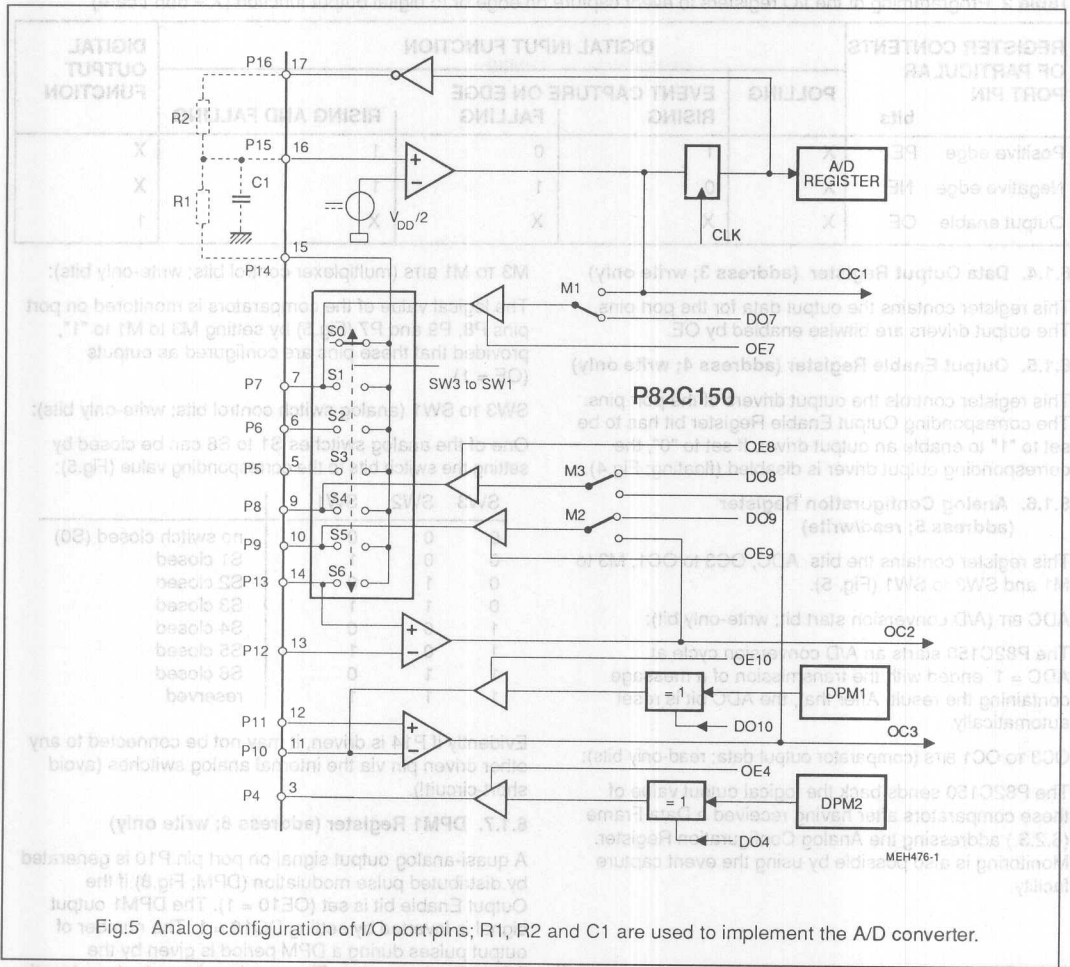
**6.1.9. A/D Register (address 8; read only)**

This register contains the result of the A/D converted level of that I/O pin which was selected by the SW bits. The conversion is started by ADC-bit set to "1", or by transmitting a Data Frame addressing the A/D Register.



CAN serial linked I/O device  
with digital and analog port functions

82C150



# CAN serial linked I/O device with digital and analog port functions

82C150

## 6.2. CAN functions

The P82C150 meets the CAN protocol specification version 2.0 A and B (passive) with restricted bit timing because of the on-chip RC-oscillator.

In a system with P82C150 nodes there must be at least one conventional crystal-operated CAN controller (host node), that is responsible to control P82C150 nodes. P82C150 nodes cannot control each other.

Each time a P82C150 node receives a Data Frame, it initiates the transmission of a Data Frame containing four bits status information, the register address (previously received) and the current contents of the addressed register. This enables the host node to verify that the addressed register has correctly been written in case of writable registers, and to read the contents in case of readable registers.

**Table 3** Message types and format

FRAME	TRANSMISSION	RECEPTION
Data Frame	yes (DLC = 3; DIR = 1)	yes (DLC = 3; DIR = 0; calibration message with DLC = 2 to 8 allowed)
Remote Frame	no	yes (DLC = 3; DIR = 1)
Error Frame	yes	yes
Overload Frame	yes (only as a response)	yes

### 6.2.1. CAN Identifier

Data and Remote Frames to be processed by the P82C150 are of Standard Format with 11 Identifier bits ID.10 to ID.0. Frames with extended Identifier (CAN specification version 2.0 B) are ignored.

ID.10	Identifier	ID.0
0 1 P3 1 0 P2 P1	P0 1 0 DIR	RTR

1 = recessive; 0 = dominant

P3 to P0 programmable identifier bits read from port pins P3 to P0 during reset.

DIR "1" for transmission of Data Frames to the host. It must be set to "1" in Remote Frames and to "0" in Data Frames received from the host.

RTR Remote Transmission Request bit.

The input levels on P3 to P0, for example set by resistors to  $V_{SS}$  or  $V_{DD}$ , are latched in the Identifier latch with the falling edge of the RST input signal. They represent the variable part of the Identifier, while the remaining bits are

fixed (mask-programmed). P3 to P0 can be used as I/O ports after reset.

The way of identifier programming is based on two facts:

- each P82C150 operates with only two Identifiers at which the higher priority Identifier is used for Data Frame reception (there is an extra Identifier for calibration purpose)
- there can be maximum sixteen P82C150 circuits in one network.

### 6.2.2. Transmission of Data Frames

Data Frames transmitted by the P82C150 contain three data bytes (Fig.6). The first data byte contains the status information and the register address A3 to A0, the other two data bytes contain the content of the addressed I/O Register.

status	register address
RSTD EW BM1 BM0	A3 A2 A1 A0

RSTD It is "1" in the first message after the successful detection of the bit rate (bit time calibrated);

EW "1", if the error warning limit (32) is reached

BM1, BM0 bus mode status bits

A3 to A0 register address bits

The EW status bit is set when the Receive Error Counter or the Transmit Error Counter have exceeded the Error Warning Limit of 32, also temporarily, since the last successful transmission of a message.

After each successful message transmission, the P82C150 delays the transmission of a possibly further pending message for three bit times. The reason is to give other CAN controllers – with a lower identifier priority – the possibility to transmit a message in case of faulty contact at one of the edge-triggered port pins.

### 6.2.3. Reception of Data Frames and Remote Frames

Received Data Frames have the same format as transmitted ones, only the DIR-bit (ID.0) in the Arbitration Field is different. The status bits RSTD, EW, BM1 and BM0 are ignored during reception.

The P82C150 confirms each reception of a Data Frame by transmitting a Data Frame containing the (new) contents of the addressed I/O Register.

## CAN serial linked I/O device with digital and analog port functions

82C150

Exceptions from this rule:

1. Analog Configuration Register:  
If a P82C150 receives a Data Frame addressing the Analog Configuration Register and the ADC bit is set to "1", it will respond with two messages. The first message returns the contents of the Analog Configuration Register. The control instructions are executed (e. g. next analog input channel selected), and an A/D conversion cycle is started after a set-up time. After finishing the A/D conversion cycle, the second message is transmitted containing the result (A/D Register).
2. A/D Register:  
On receiving a Data Frame addressing the A/D Register, the P82C150 starts an A/D conversion cycle. It automatically returns the result of the conversion (A/D Register) by transmitting a respective Data Frame after finishing the A/D conversion cycle.
3. At normal operation, the calibration messages are confirmed by returning a dominant bit in the acknowledge slot. There is no particular confirmation message returned by the P82C150. Only after entering the calibrated state (start-up), a Data Frame containing the Data Input Register contents is transmitted indicating to the host node, that the P82C150 is now ready for transmission.

Remote Frame:

Received Remote Frames must have the Data Length Code DLC = 3 (Remote Frames with DLC ≠ 3 are ignored). It is answered by a Data Frame containing the contents of the Data Input Register.

### 6.2.4. CAN-bus modes

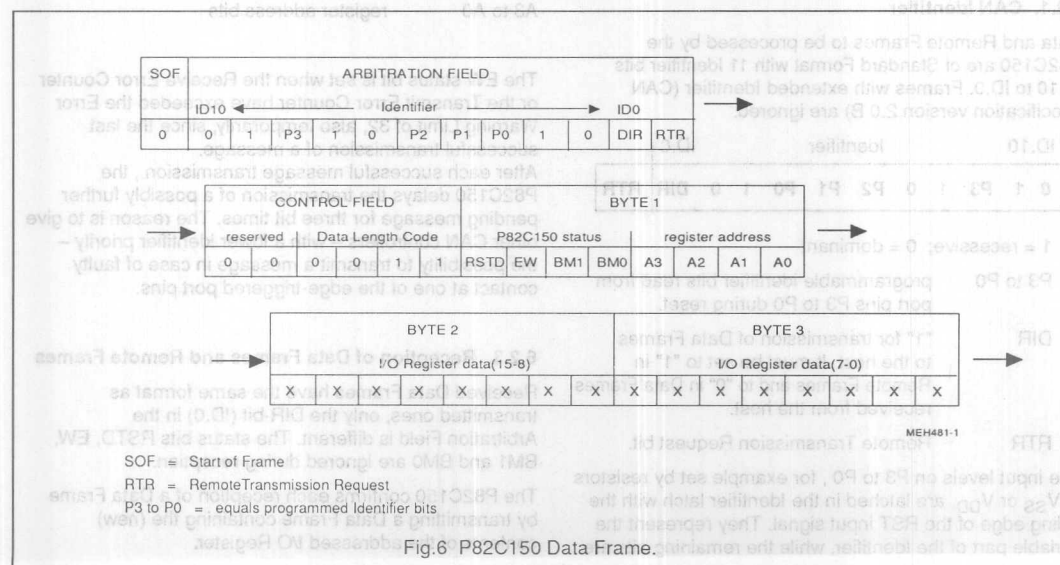
The P82C150 can pass through four CAN-bus modes under certain conditions (Fig.7). In the bus modes 0 to 2 (Table 4) the P82C150 is operating with different input comparator configurations. Bus mode 3 is the power reduced Sleep Mode.

The bus modes support:

- communication on two balanced wires (differential system)
- communication on one wire in a two-wire differential system
- Sleep Mode with wake-up via either of the two RX inputs
- connection of a second transmission medium (redundancy)

There are two possibilities for condition 1 to switch to the next mode:

- overflow of the bit counter when 8192 is reached since the last calibration message or since end of reset





# CAN serial linked I/O device with digital and analog port functions

82C150

Table 4 CAN-bus modes

BUS MODE	BITS BM1, BM0	RECEPTION LEVEL RECESSIVE	DOMINANT	TRANSMISSION	
				TX1	TX0
0 Differential	00	RX0 > RX1	RX0 < RX1	enabled	enabled
1 One-wire RX1	01	RX1 < REF	RX1 > REF	enabled	enabled
2 One-wire RX0	10	RX0 > REF	RX0 < REF	disabled	enabled
3 Sleep	11	RX0 > REF and RX1 < REF	RX0 < REF or RX1 > REF	disabled	disabled

**Note to Table 4**

Output TX1 is disabled in bus mode 2 to tolerate short-circuit between the CAN-bus wires CAN\_H and CAN\_L.

overflow of the Transmit Error Counter  
(>255; bus-off limit reached).

The P82C150 is in mode 0 after reset. All I/O Registers are cleared (outputs floating) when bus mode changes (e.g. by bus-off or Sleep Mode); the status bits are set additionally:

RSTD = 1  
EW = 0  
BM<sub>new</sub> = BM<sub>old</sub> + 1

The programmed Identifier bits remain unchanged.

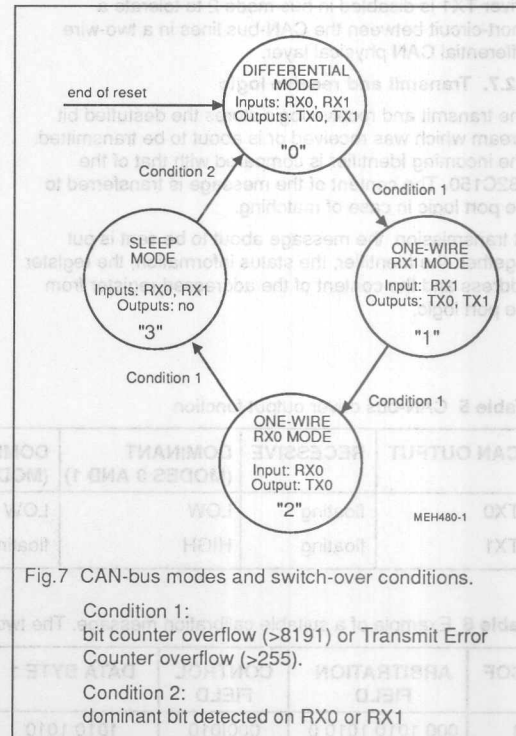
During Sleep Mode, the internal RC oscillator is stopped, and all the output drivers are disabled (I/O Register contents cleared). A P82C150 in Sleep Mode can only be awoken via CAN-bus (dominant level on RX0 or RX1) or by a reset condition:

**6.2.5. Bit timing**

The Nominal Bit Time of the P82C150 is subdivided into 10 Time Quanta. The Synchronization Time Segment (SYNC\_SEG) and the Propagation Time Segment (PROP\_SEG) are each one Time Quantum long. The Phase Buffer Segment 1 (PHASE\_SEG1) and the Phase Buffer Segment 2 (PHASE\_SEG2) are each four Time Quanta long. The Resynchronization Jump Width (SJW) is four Time Quanta long. The sample point is located at the end of the Phase Buffer Segment 1. The Nominal Bit Time is internally adjusted to that bit timing which is provided by the crystal driven host (calibration message).

1 BIT TIME									
BT1	BT2	BT3	BT4	BT5	BT6	BT7	BT8	BT9	BT10
SYNC_SEG	PROP_SEG	PHASE_SEG1				PHASE_SEG2			

The usable bus length at a given bit rate is reduced in comparison to other CAN controllers with programmable bit timing because the Propagation Time Segment is fixed to 1/10 length of the Nominal Bit Time.



## CAN serial linked I/O device with digital and analog port functions

82C150

### 6.2.6. CAN-bus transceiver

The transceiver of the P82C150 consists of the configurable input comparator and of complementary open-drain driver outputs. The reference voltage REF is an additional output.

#### CAN-BUS INPUT COMPARATOR (RX0, RX1):

The input comparator monitors the transient voltage on RX1 and RX0.

The result of the input comparator is "1" if the voltage levels of the CAN-bus lines are regarded as recessive, and "0" if they are regarded as dominant.

The recessive state and the dominant state are not equivalent and may not be mixed-up.

The input comparator is configurable depending on the four CAN-bus modes (Table 4), supporting battery-powered applications (Sleep Mode) and tolerance against bus wiring failures.

#### CAN-BUS OUTPUT DRIVERS (TX0, TX1)

The output driver function is shown in Table 5. The output driver TX1 is disabled in bus mode 2 to tolerate a short-circuit between the CAN-bus lines in a two-wire differential CAN physical layer.

### 6.2.7. Transmit and receive logic

The transmit and receive logic stores the destuffed bit stream which was received or is about to be transmitted. The incoming Identifier is compared with that of the P82C150. The content of the message is transferred to the port logic in case of matching.

At transmission, the message about to be sent is put together: the Identifier, the status information, the register address and the content of the addressed register from the port logic.

### 6.2.8. Bit Stream Processor and Error Management Logic

The Bit Stream Processor (BSP) is a sequencer to control the data stream between the transmit/receive logic (parallel data) and the on-chip CAN transceiver (serial data). Reception/transmission, bit stuffing/destuffing, arbitration and error detection, according to CAN protocol specification version 2.0 A and B (passive), are performed. Further, automatic re-transmission of corrupted messages is handled by means of continuously comparing the output bit stream with the input bit stream.

Moreover, the Bit Stream Processor provides control information to calibrate the internal bit time.

The Error Management Logic is responsible for the complete CAN-inherent error management.

### 6.2.9. Oscillator and calibration

The P82C150 contains an on-chip RC-oscillator. The bit time is automatically calibrated by messages being received via CAN-bus. During start-up (after wake-up or reset) any message is used to calibrate the bit time until the calibration is sufficient to receive messages correctly. From this time on, the bit time is calibrated and fine-tuned by calibration messages with a special Identifier transmitted by the crystal-operated host.

Only P82C150 nodes being calibrated by calibration messages can transmit messages. The first message is transmitted directly after entering the calibrated state.

Since the P82C150 is not able to transmit as long as the bit time is not calibrated, it cannot wake-up other CAN nodes via the bus line. Hence to keep the network alive, the calibration message must be transmitted regularly by a crystal-operated (host) node with a maximum repetition period of 8192 bit. It is recommended to transmit two calibration messages within a period of 8192 bit.

Table 5 CAN-bus driver output function

CAN OUTPUT	RECESSIVE	DOMINANT (MODES 0 AND 1)	DOMINANT (MODE2)	RESET STATE, BUS-OFF AND SLEEP MODE (MODE 3)
TX0	floating	LOW	LOW	floating
TX1	floating	HIGH	floating	floating

Table 6 Example of a suitable calibration message. The two important I/O transitions are marked (1).

SOF	ARBITRATION FIELD	CONTROL FIELD	DATA BYTE 1	DATA BYTE 2	CRC FIELD
0	000 1010 1010 0	0001010	1010 1010	00001 0100	00010 1011 1000 0010

1 = stuff bit (recessive); (1) the total length is 67 bit from start-of -frame to end-of-intermission.

## CAN serial linked I/O device with digital and analog port functions

82C150

### 6.2.10. Calibration message

The calibration message has to meet the following requirements

- transmitted by a crystal-operated node (host node)
- identifier: 000 1010 1010 (1 = recessive; 0 = dominant)
- RTR bit: 0
- allowed control field: DLC = 2 to 8
- the first recessive to dominant transition after the control field must be followed by another recessive to dominant transition in a distance of exactly 32 bit (stuff bits included).

Example of a suitable calibration message (there are others using different data bytes; Table 6):

data length code: 0010  
1st data byte 1010 1010 (AAh)  
2nd data byte 0000 0100 (04h)

### 6.3. Initialization

#### 6.3.1. Identifier programming

Most of the P82C150 identifier bits are fixed. Four bits are programmable via port pins P3 to P0. All output drivers are disabled at reset, also P3 to P0. Thus the outputs are floating unless the input level is defined by external components to define identifier bits. They are latched at the end of reset, and P3 to P0 can be used as port pins. It is not allowed, according to the CAN protocol specification, that multiple bus nodes transmit the same identifier bit combination. Therefore a P82C150 must have one of the 16 possible identifier bit combinations, one that is not yet occupied.

#### 6.3.2. Reset function

RST = HIGH disables all output drivers P16 to P0, TX0 and TX1. All I/O Registers are automatically cleared and set to "0". The bit time is set greater than 50  $\mu$ s.

After reset:

status bits	identifier bits
RSTD = 1	ID.8 equals P3
EW = 0	ID.5 equals P2
BM1 = 0	ID.4 equals P1
BM0 = 0	ID.3 equals P0

If a particular clock period is necessary, e. g. for a dedicated DPM output frequency, this can be achieved by feeding an external clock signal into P0. RST and TEST must be permanently HIGH for this special mode. A reset is then performed as usual (RST = HIGH; TEST = LOW).

### 6.3.3. Bit time calibration

The P82C150 must receive at least two messages to calibrate its bit time after reset or change of bus mode. The first message is used to detect the bit time length between two consecutive falling edges at the output of the CAN input comparator. Therefore the bit stream should contain a sequence of "1010".

If this measurement is not disturbed, the P82C150 is able to receive messages correctly and to execute the respective commands. The P82C150 is fully calibrated and transmits its first bit (acknowledge) after a successful reception of a calibration message. Thus the P82C150 is enabled to transmit messages.

Sign-on message:

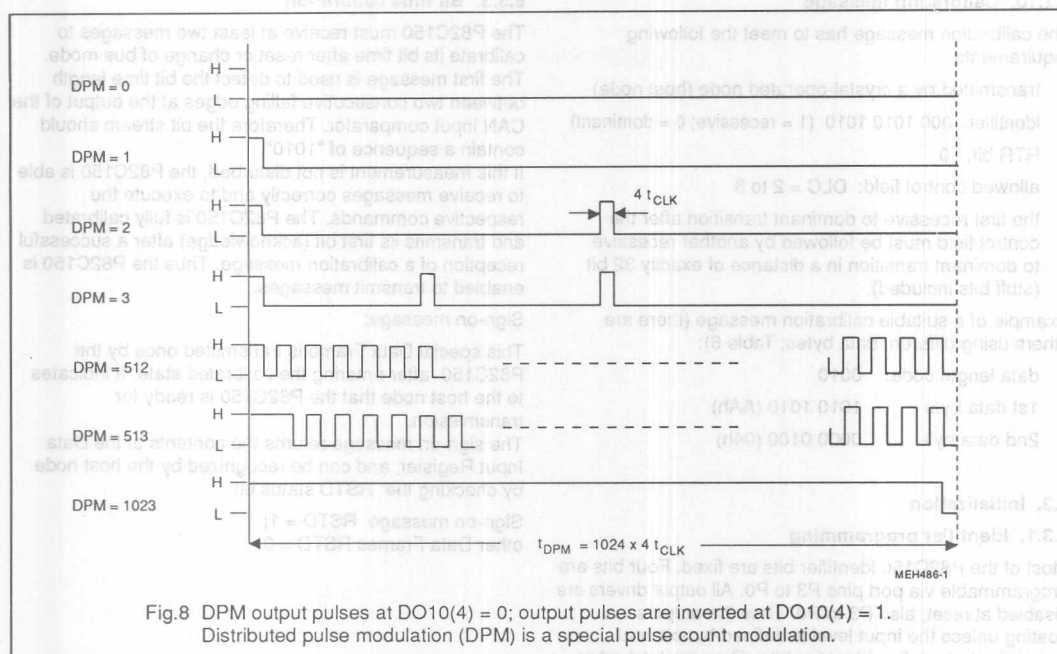
This special Data Frame is transmitted once by the P82C150 after entering the calibrated state. It indicates to the host node that the P82C150 is ready for transmission.

The sign-on message returns the contents of the Data Input Register, and can be recognized by the host node by checking the RSTD status bit.

Sign-on message RSTD = 1;  
other Data Frames RSTD = 0.

# CAN serial linked I/O device with digital and analog port functions

82C150



## 7. LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134).

SYMBOL	PARAMETER	MIN.	MAX.	UNIT
$V_{DD}$	supply voltage on $V_{DD}$ pins	-0.5	6.5	V
$V_I$	DC input voltage on any pin (RX0, RX1, TX0, TX1 excluded)	-0.5	$V_{DD}+0.5$	V
$I_I$	RX1 and RX0 input current	-	$\pm 2$	mA
$I_{REF}$	reference output current	-	$\pm 2$	mA
$I_O$	port output current at port enabled (pins P0 to P15)	-	$\pm 5$	mA
	port output current at analog switch enabled (OE-bits = 0; pins P5 to P9, P13, P14)	-	$\pm 7.5$	mA
	TX0 and TX1 output current	-	$\pm 30$	mA
$P_{O\ tot}$	total output power dissipation (port outputs together)	-	200	mW
$T_{amb}$	operating ambient temperature range	-40	+125	°C
$T_{stg}$	storage temperature range	-65	+150	°C
$P_{tot}$	total power dissipation	-	1	W

# CAN serial linked I/O device with digital and analog port functions

82C150

## 8. CHARACTERISTICS

### 8.1. DC CHARACTERISTICS

$V_{DD} = 5\text{ V} \pm 4\%$ ;  $V_{SS} = 0\text{ V}$  and  $T_{amb} = -40$  to  $+125\text{ }^{\circ}\text{C}$  unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
$V_{DD}$	supply voltage range		4.8	5.2	V
$I_{DD}$	supply current	$V_{RST} = V_{DD}$ ; all outputs unloaded	-	22	mA
$I_{SM}$	Sleep Mode supply current	all outputs unloaded	-	tbf	$\mu\text{A}$
<b>CAN input comparator RX0 and RX1</b> $1.5\text{ V} < V_I < (V_{DD2}-1.5\text{ V})$					
$V_{DIF}$	differential input voltage		$\pm 25$	-	mV
$V_{HYST}$	input voltage hysteresis	note 1	8	-	mV
$I_I$	input current		-	$\pm 400$	nA
<b>CAN output driver TX0 and TX1</b> port pins P16 to P0 unloaded					
$V_{OLT}$	output voltage LOW	$I_{OLT} = 1.5\text{ mA}$	-	0.1	V
$V_{OHT}$	output voltage HIGH	$I_{OHT} = -1.5\text{ mA}$	$V_{DD}-0.1$	-	V
<b>Reference voltage REF</b>					
$V_{REF}$	reference output voltage	$I_o < \pm 75\text{ }\mu\text{A}$			V
	minimum		$1/2 V_{DD2}-\text{tbf}$		mV
	maximum			$1/2 V_{DD2}+\text{tbf}$	mV
<b>Control inputs RST and TEST</b>					
$V_{IL}$	input voltage LOW		-0.5	1.5	V
$V_{IH}$	input voltage HIGH		3.5	$V_{DD}+0.5$	V
$V_{HYST}$	input voltage hysteresis	note 1	0.5	-	V
<b>Digital port inputs P15 to P0</b>					
$V_{IL}$	input voltage LOW		-0.5	1.5	V
$V_{IH}$	input voltage HIGH		3.5	$V_{DD}+0.5$	V
<b>Digital port outputs P16 to P0</b> OE bits set					
$V_{OL}$	output voltage LOW	$I_{OL} = 4\text{ mA}$ (sink)	-	1	V
$V_{OH}$	output voltage HIGH	$I_{OH} = -4\text{ mA}$ (source)	$V_{DD}-1$	-	V
<b>OC2 comparator P12, P13 and OC3 comparator P10, P11</b>					
$V_{DIF1}$	differential input voltage	$1.5\text{ V} < V_I < (V_{DD}-1.5)\text{ V}$	tbf	-	mV



# CAN serial linked I/O device with digital and analog port functions

82C150

## 9. APPLICATION INFORMATION

### Maximum bus length

**Table 7** Maximum bus line length for CAN-bus systems with P82C150 nodes. The bit timing parameters refer to using a P8xCE598 or P8xC592 microcontroller with on-chip CAN interface as a host node (Fig.12).

Assumptions:

- the total in/out delay of external transceiver circuit is 120 ns (e. g. PCA82C250 CAN transceiver; Fig.13)
- the propagation delay on the transmission medium is 5.5 ns/m.

BIT RATE (kbit/s)	$t_{prop} / 2$ (note1) ( $\mu$ s)	MAXIMUM BUS LENGTH	BIT TIMING (P8x CE 598/P8xC592)		
			$f_{CLK}$ (MHz)	BTR0 (note 2)	BTR1 (note 2)
125	0.8	80 m	15	C5 h	34 h
100	1	120 m	16	C7 h	34 h
50	2	300 m	16	CE h	34 h
20	5	850 m	16	E7 h	34 h

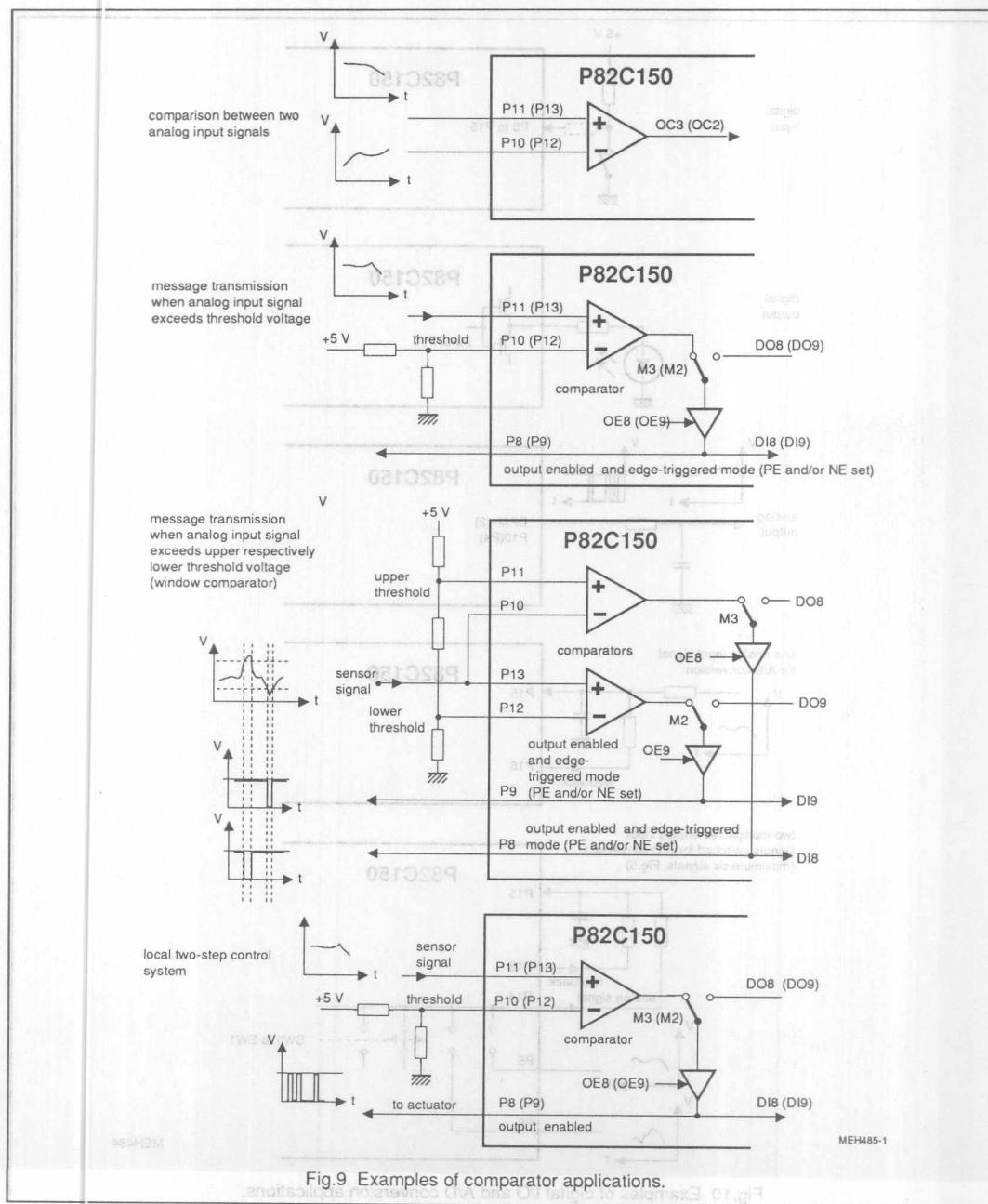
### Notes to Table 7

- $t_{prop} / 2$  is the maximum propagation delay between two CAN-bus nodes (delays of on- and off-chip transceiver circuits included)
- BTR0 and BTR1 (hex values) are particular configuration registers referring to bit timing.

V	-	$V_{DD} - 0.1$	$I_{OH} = -1.8$ mA	output voltage HIGH	V <sub>OH</sub>
V	-	$V_{DD} - 0.1$	$I_{OL} = 1.8$ mA	output voltage LOW	V <sub>OL</sub>
V <sub>REF</sub>	-	-	$I_{REF} < 275$ $\mu$ A	reference output voltage	V <sub>REF</sub>
V <sub>m</sub>	-	$V_{DD} - 0.5$	-	minimum	V <sub>m</sub>
V <sub>m</sub>	-	$V_{DD} + 0.5$	-	maximum	V <sub>m</sub>
Control inputs RST and TEST					
V <sub>L</sub>	-	-0.5	-	input voltage LOW	V <sub>L</sub>
V <sub>H</sub>	-	0.5	-	input voltage HIGH	V <sub>H</sub>
V <sub>HYST</sub>	-	0.5	-	input voltage hysteresis (note 1)	V <sub>HYST</sub>
Digital port inputs P15 to P0					
V <sub>L</sub>	-	-0.5	-	input voltage LOW	V <sub>L</sub>
V <sub>H</sub>	-	0.5	-	input voltage HIGH	V <sub>H</sub>
Digital port outputs P15 to P0					
V <sub>L</sub>	-	-	$I_{OL} = 4$ mA (sink)	output voltage LOW	V <sub>L</sub>
V <sub>H</sub>	-	$V_{DD} - 1$	$I_{OH} = 4$ mA (source)	output voltage HIGH	V <sub>H</sub>
OCS comparator P15, P13 and OCS comparator P10, P11					
V <sub>m</sub>	-	-	$1.5$ V $< V_i < V_{DD} - 1.5$ V	differential input voltage	V <sub>m</sub>

# CAN serial linked I/O device with digital and analog port functions

82C150



# CAN serial linked I/O device with digital and analog port functions

82C150

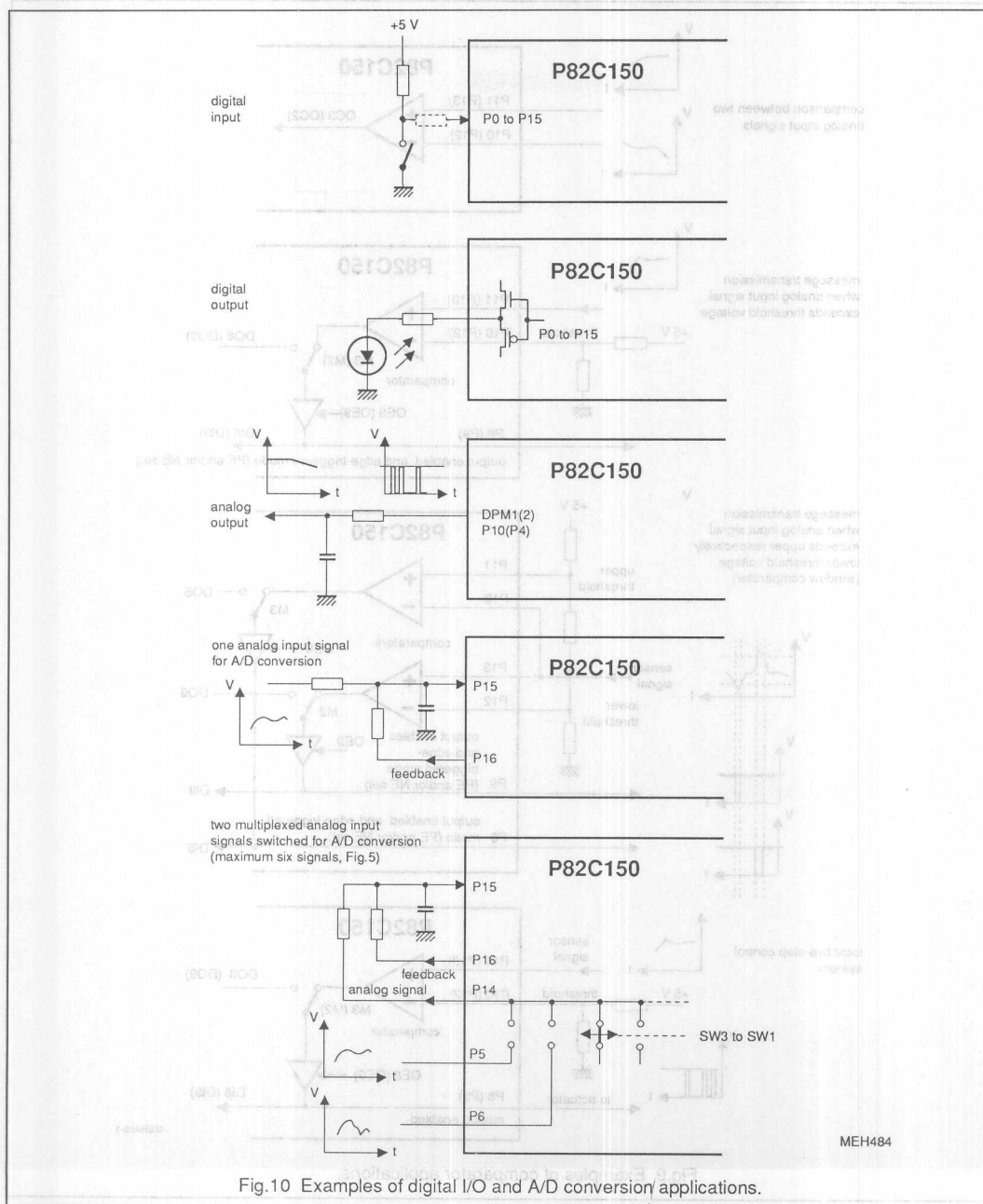


Fig.10 Examples of digital I/O and A/D conversion applications.

# CAN serial linked I/O device with digital and analog port functions

82C150

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
<b>A/D comparator input P15</b> $1.5\text{ V} < V_I < (V_{DD2}-1.5)\text{ V}$					
$V_{i\text{sw}}$	input switch-over voltage		$V_{DD}/2 - \text{tbf}$	-	V
	minimum		-	$V_{DD}/2 + \text{tbf}$	V
$I_{LI}$	input leakage current		-	$\pm 1$	$\mu\text{A}$
<b>Analog switches</b> $I_{ON} = \pm 4\text{ mA}$					
$R_{ON}$	ON resistance	between P5 to P9, P13 and P14; note 1	-	tbf	$\Omega$

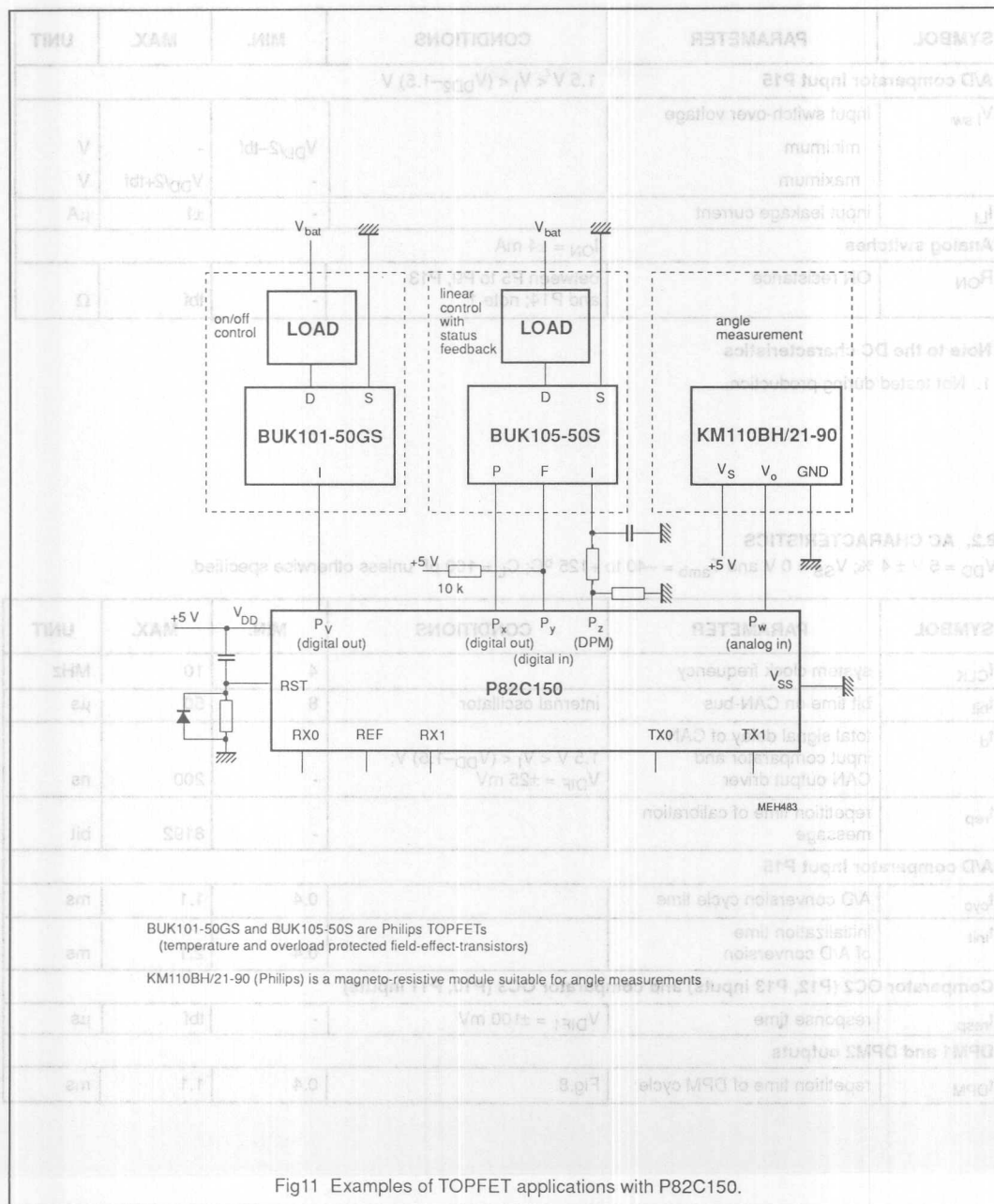
## Note to the DC characteristics

- Not tested during production.

## 8.2. AC CHARACTERISTICS

$V_{DD} = 5\text{ V} \pm 4\%$ ;  $V_{SS} = 0\text{ V}$  and  $T_{\text{amb}} = -40$  to  $+125\text{ }^\circ\text{C}$ ;  $C_L = 100\text{ pF}$  unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
$f_{\text{CLK}}$	system clock frequency		4	10	MHz
$t_{\text{bit}}$	bit time on CAN-bus	internal oscillator	8	50	$\mu\text{s}$
$t_d$	total signal delay of CAN input comparator and CAN output driver	$1.5\text{ V} < V_I < (V_{DD}-1.5)\text{ V}$ ; $V_{DIF} = \pm 25\text{ mV}$	-	200	ns
$t_{\text{rep}}$	repetition time of calibration message		-	8192	bit
<b>A/D comparator input P15</b>					
$t_{\text{cyc}}$	A/D conversion cycle time		0.4	1.1	ms
$t_{\text{init}}$	initialization time of A/D conversion		0.4	2.1	ms
<b>Comparator OC2 (P12, P13 inputs) and comparator OC3 (P10, P11 inputs)</b>					
$t_{\text{resp}}$	response time	$V_{DIF1} = \pm 100\text{ mV}$	-	tbf	$\mu\text{s}$
<b>DPM1 and DPM2 outputs</b>					
$t_{\text{DPM}}$	repetition time of DPM cycle	Fig.8	0.4	1.1	ms





# CAN serial linked I/O device with digital and analog port functions

82C150

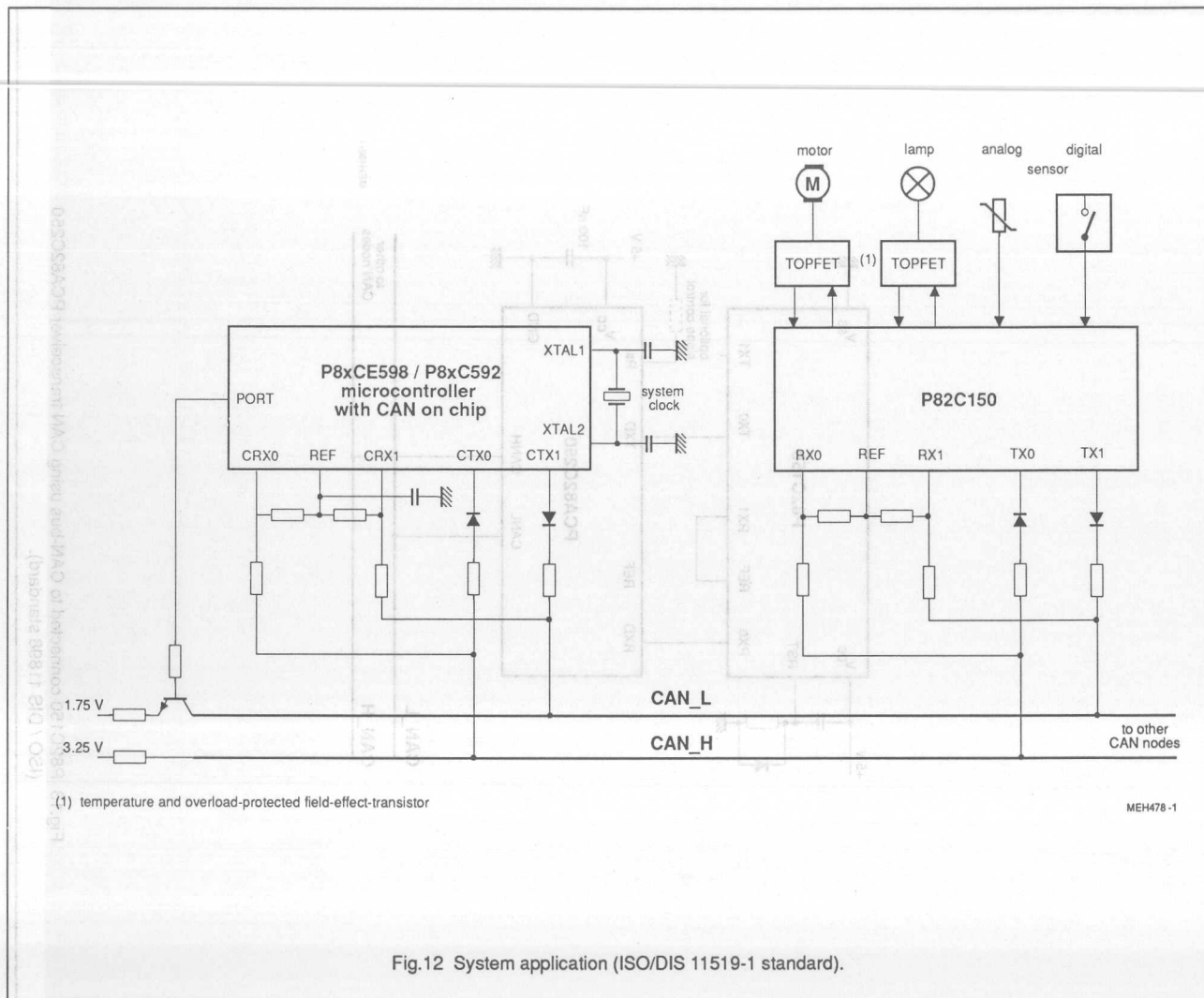


Fig.12 System application (ISO/DIS 11519-1 standard).

# CAN serial linked I/O device with digital and analog port functions

82C150

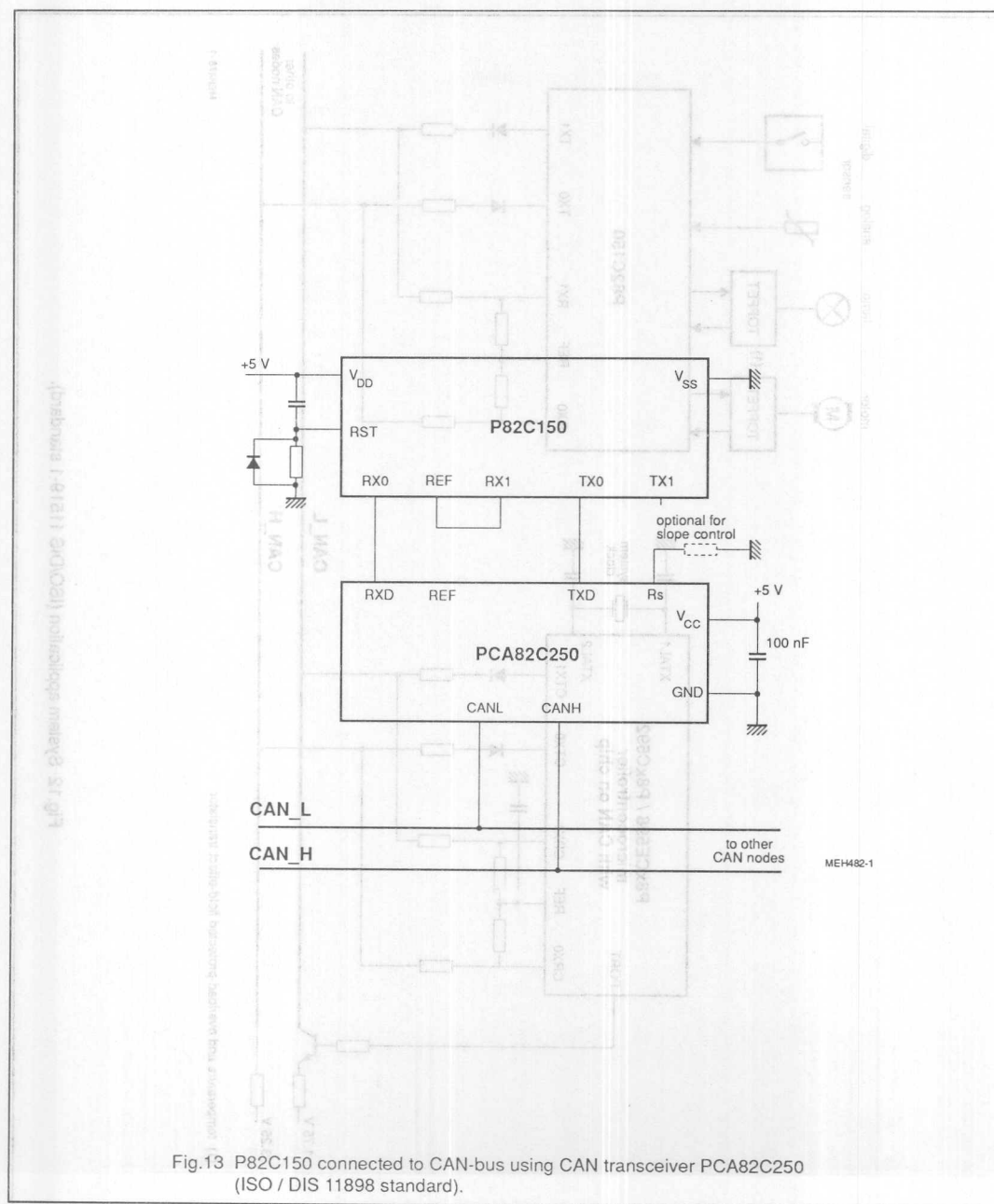


Fig.13 P82C150 connected to CAN-bus using CAN transceiver PCA82C250 (ISO / DIS 11898 standard).

## Stand-alone CAN-controller

## 82C200

## 1 FEATURES

- Multi-master architecture
- Interfaces with a large variety of microcontrollers
- Bus access priority (determined by the message identifier)
- 2032 message identifiers
- Guaranteed latency time for high priority messages
- Powerful error handling capability
- Data length from 0 to 8 bytes
- Configurable bus interface
- Programmable clock output
- Multicast and Broadcast message facility
- Non destructive bit-wise arbitration
- Non-return-to-zero (NRZ) coding/decoding with bit-stuffing
- Programmable transfer rate (up to 1 Mbit/s)
- Programmable output driver configuration
- Suitable for use in a wide range of networks including the SAE networks Class A, B and C
- 16 MHz clock frequency
- -40 to +85/125 °C operating temperature.

## 2 GENERAL DESCRIPTION

The PCA82C200, PCF82C200 (hereafter generically referred to as PCX82C200) is a highly integrated stand-alone controller for the controller area network (CAN) used within automotive and general industrial environments. The temperature range includes an automotive temperature range version (PCA82C200) of -40 to +125 °C and a -40 to +85 °C version (PCF82C200) for general applications.

The PCX82C200 contains all necessary features required to implement a high performance communication protocol. The PCX82C200 with a simple bus line connection performs all the functions of the physical and data-link layers. The application layer of an Electronic Control Unit (ECU) is provided by a microcontroller, to which the PCX82C200 provides a versatile interface. The use of the PCX82C200 in an automotive or general industrial environment, results in a reduced wiring harness and an enhanced diagnostic and supervisory capability.

## 3 ORDERING AND PACKAGE INFORMATION

PHILIPS PART ORDER NUMBER PART MARKING	PHILIPS NORTH AMERICA PART ORDER NUMBER <sup>1</sup>	PACKAGE				TEMPERATURE RANGE (°C)
		PINS	PIN POSITION	MATERIAL	CODE	
PCA82C200P	PCA82C200PN	28	DIL	plastic	SOT117	-40 to +125
PCA82C200T	PCA82C200TD	28	SO28	plastic	SOT136A	-40 to +125
PCF82C200P	PCF82C200PN	28	DIL	plastic	SOT117	-40 to +85
PCF82C200T	PCF82C200TD	28	SO28	plastic	SOT136A	-40 to +85

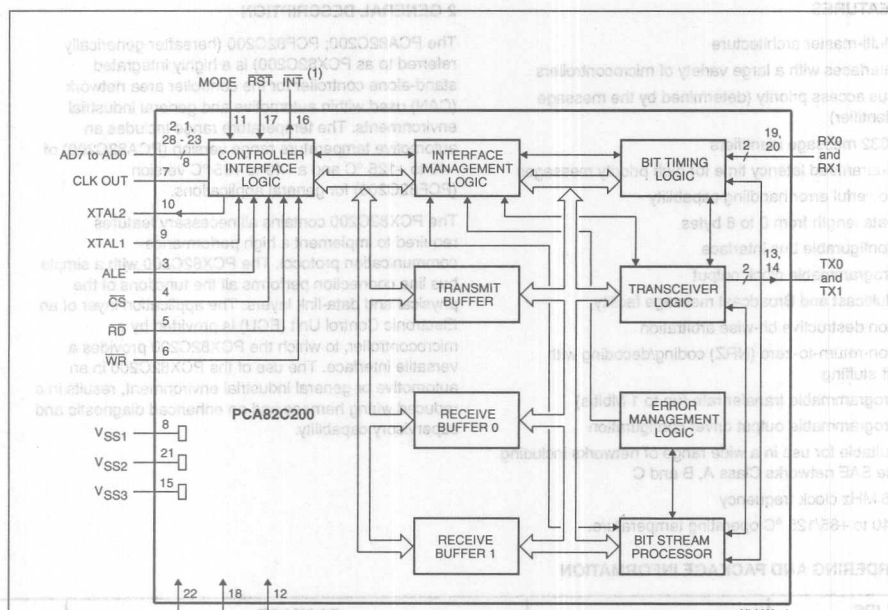
## NOTE:

1. Parts ordered by the Philips North America part number will be marked with the Philips part marking.

TEMPERATURE RANGE (°C)	CODE	MATERIAL	PIN POSITION	PINS	PHILIPS NORTH AMERICA	
					PART ORDER NUMBER	PART MARKING
-40 to +125	SO18	plastic	28	28	PCA82C200N	PCA82C200
-40 to +125	SO18	plastic	28	28	PCA82C200D	PCA82C200
-40 to +85	SO18	plastic	28	28	PCA82C200N	PCA82C200
-40 to +85	SO18	plastic	28	28	PCA82C200D	PCA82C200

(1) During 'sleep mode' this pin functions as an input.

Fig.1 Block diagram.



## Stand-alone CAN-controller

## 82C200

#### 4 PINNING

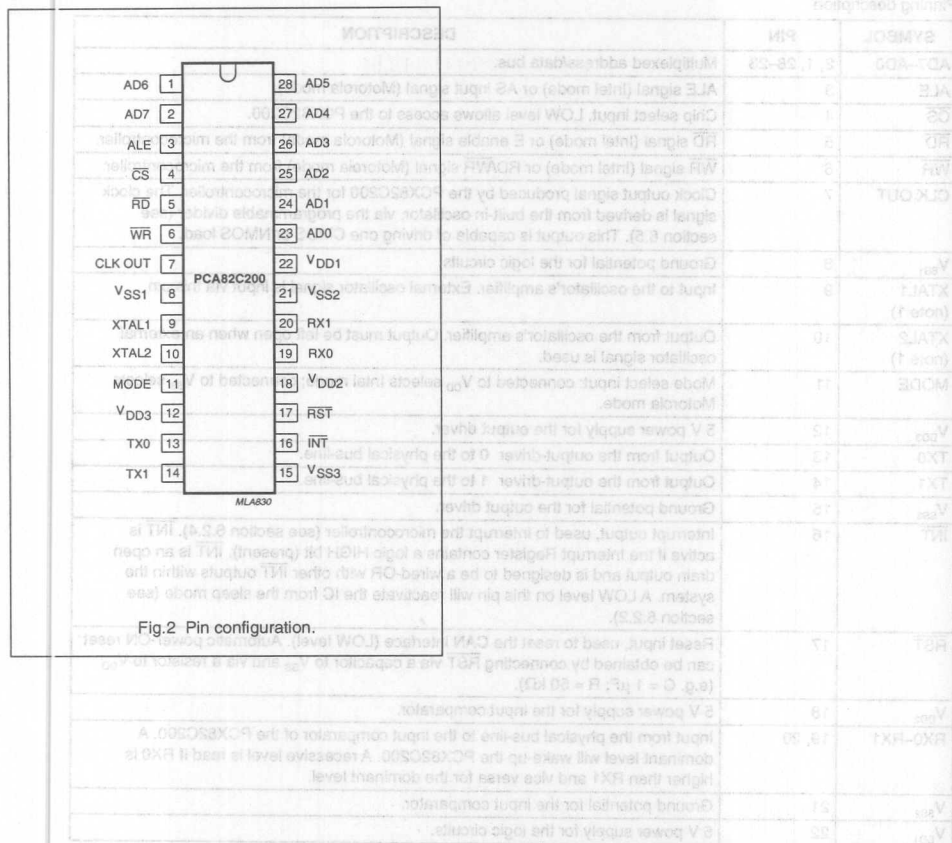


Fig.2 Pin configuration.



## Stand-alone CAN-controller

## PCX82C200

## Pinning description

SYMBOL	PIN	DESCRIPTION
AD7-AD0	2, 1, 28-23	Multiplexed address/data bus.
ALE	3	ALE signal (Intel mode) or AS input signal (Motorola mode).
CS	4	Chip select input, LOW level allows access to the PCX82C200.
RD	5	RD signal (Intel mode) or E enable signal (Motorola mode) from the microcontroller.
WR	6	WR signal (Intel mode) or RD/WR signal (Motorola mode) from the microcontroller.
CLK OUT	7	Clock output signal produced by the PCX82C200 for the microcontroller. The clock signal is derived from the built-in oscillator, via the programmable divider (see section 6.5). This output is capable of driving one CMOS or NMOS load.
V <sub>SS1</sub>	8	Ground potential for the logic circuits.
XTAL1 (note 1)	9	Input to the oscillator's amplifier. External oscillator signal is input via this pin.
XTAL2 (note 1)	10	Output from the oscillator's amplifier. Output must be left open when an external oscillator signal is used.
MODE	11	Mode select input: connected to V <sub>DD</sub> selects Intel mode; connected to V <sub>SS</sub> selects Motorola mode.
V <sub>DD3</sub>	12	5 V power supply for the output driver.
TX0	13	Output from the output-driver 0 to the physical bus-line.
TX1	14	Output from the output-driver 1 to the physical bus-line.
V <sub>SS3</sub>	15	Ground potential for the output driver.
INT	16	Interrupt output, used to interrupt the microcontroller (see section 6.2.4). INT is active if the Interrupt Register contains a logic HIGH bit (present). INT is an open drain output and is designed to be a wired-OR with other INT outputs within the system. A LOW level on this pin will reactivate the IC from the sleep mode (see section 6.2.2).
RST	17	Reset input, used to reset the CAN interface (LOW level). Automatic power-ON reset can be obtained by connecting RST via a capacitor to V <sub>SS</sub> and via a resistor to V <sub>DD</sub> (e.g. C = 1 µF; R = 50 kΩ).
V <sub>DD2</sub>	18	5 V power supply for the input comparator.
RX0-RX1	19, 20	Input from the physical bus-line to the input comparator of the PCX82C200. A dominant level will wake-up the PCX82C200. A recessive level is read if RX0 is higher than RX1 and vice versa for the dominant level.
V <sub>SS2</sub>	21	Ground potential for the input comparator.
V <sub>DD1</sub>	22	5 V power supply for the logic circuits.

## Note

1. XTAL1 and XTAL2 pins should be connected to V<sub>SS</sub> via 15 pF capacitors.

## 5 FUNCTIONAL DESCRIPTION

The PCX82C200 contains all necessary hardware for a high performance serial network communication (see Fig.1). The PCX82C200 controls the communication flow through the area network using the CAN-protocol. The PCX82C200 meets the following automotive requirements:

- short message length
- guaranteed latency time for urgent messages
- bus access priority, determined by the message identifier
- powerful error handling capability
- configuration flexibility to allow area network expansion.

The latency time defines the period between the initiation (Transmission Request) and the start of the transmission on the bus. Latency time is dependent on a variety of bus related conditions. In the case of a message being transmitted on the bus and one distortion the latency time can be up to 149 bit times (worst case). For more information see section 7.

### 5.1 Interface Management Logic (IML)

The IML interprets commands from the microcontroller, allocates the message buffers (TBF, RBF0 and RBF1) and provides interrupts and status information to the microcontroller.

### 5.2 Transmit Buffer (TBF)

The TBF is a 10 byte memory into which the microcontroller writes messages which are to be transmitted over the CAN network.

### 5.3 Receive Buffers (RBF0 AND RBF1)

The RBF0 and RBF1 are each 10 byte memories which are alternatively used to store messages received from the CAN network. The CPU can process one message while another is being received.

### 5.4 Bit Stream Processor (BSP)

The BSP is a sequencer, controlling the data stream between the Transmit Buffer, the Receive Buffer (parallel data) and the CAN-bus (serial data).

### 5.5 Bit Timing Logic (BTL)

The BTL synchronizes the PCX82C200 to the bitstream on the CAN-bus.

### 5.6 Transceiver Control Logic (TCL)

The TCL controls the output driver.

### 5.7 Error Management Logic (EML)

The EML performs the error confinement according to the CAN-protocol.

### 5.8 Controller Interface Logic (CIL)

The CIL is the interface to the external microcontroller. The PCX82C200 can directly interface with a variety of microcontrollers.

## 6 CONTROL SEGMENT AND MESSAGE BUFFER DESCRIPTION

The PCX82C200 appears to a microcontroller as a memory-mapped I/O device due to the on-chip RAM, guaranteeing the independent operation of both devices.

### 6.1 Address allocation

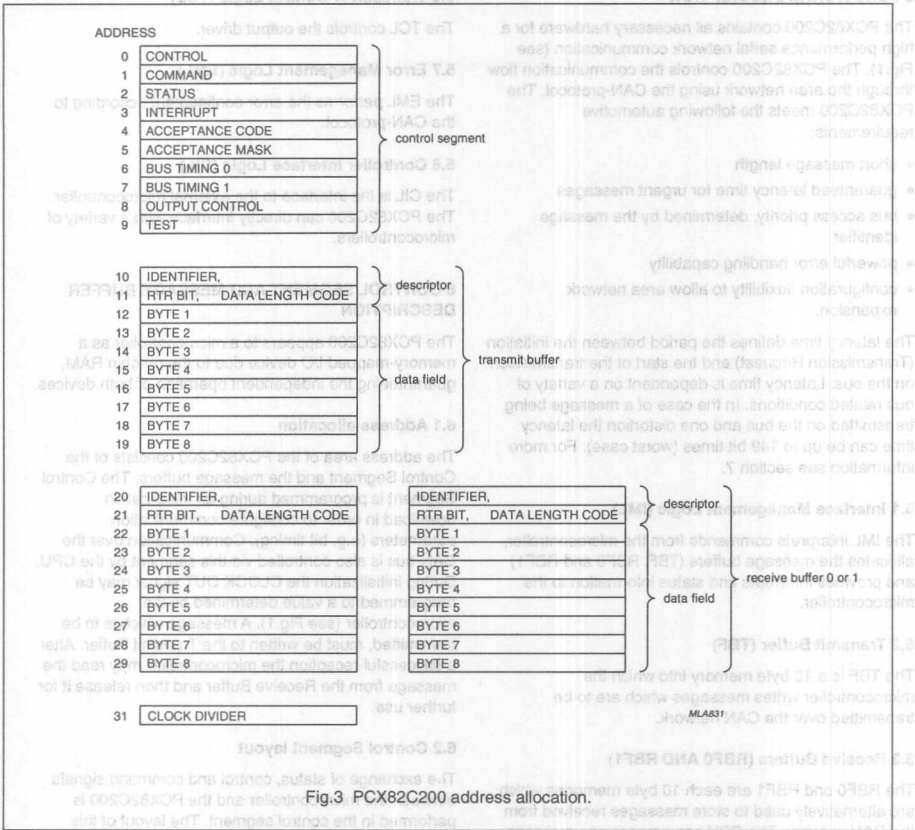
The address area of the PCX82C200 consists of the Control Segment and the message buffers. The Control Segment is programmed during an initialization download in order to configure communication parameters (e.g. bit timing). Communication over the CAN-bus is also controlled via this segment by the CPU. During initialization the CLOCK OUT signal may be programmed to a value determined by the microcontroller (see Fig.1). A message which is to be transmitted, must be written to the Transmit Buffer. After a successful reception the microcontroller may read the message from the Receive Buffer and then release it for further use.

### 6.2 Control Segment layout

The exchange of status, control and command signals between the microcontroller and the PCX82C200 is performed in the control segment. The layout of this segment is shown in Fig.3. After an initial down-load, the contents of the registers Acceptance Code, Acceptance Mask, Bus Timing Registers 0 and 1, and Output Control should not be changed. These registers may only be accessed when the Reset Request bit in the Control Register, is set HIGH (see section 6.2.1).

Stand-alone CAN-controller

82C200 CAN-controller



## Stand-alone CAN-controller

## 82C200

Table 1 Register map

TITLE	ADDR	7	6	5	4	3	2	1	0
<b>Control Segment</b>									
Control Register	0	Test Mode	Sync	reserved	Overrun Interrupt Enable	Error Interrupt Enable	Transmit Interrupt Enable	Receive Interrupt Enable	Reset Request
Command Register	1	reserved	reserved	reserved	Goto Sleep	Clear Overrun Status	Release Receive Buffer	Abort Trans-mission	Trans-mission Request
Status Register	2	Bus Status	Error Status	Transmit Status	Receive Status	Trans-mission Complete Status	Transmit Buffer Access	Data Overrun	Receive Buffer Stat
Interrupt Register	3	reserved	reserved	reserved	Wake-Up Interrupt	Overrun Interrupt	Error Interrupt	Transmit Interrupt	Receive Interrupt
Acceptance Code Register	4	AC.7	AC.6	AC.5	AC.4	AC.3	AC.2	AC.1	AC.0
Acceptance Mask Register	5	AM.7	AM.6	AM.5	AM.4	AM.3	AM.2	AM.1	AM.0
Bus Timing Register 0	6	SJW.1	SJW.0	BRP.5	BRP.4	BRP.3	BRP.2	BRP.1	BRP.0
Bus Timing Register 1	7	SAM	TSEG2.2	TSEG2.1	TSEG2.0	TSEG1.3	TSEG1.2	TSEG1.1	TSEG1.0
Output Control Register	8	OCTP1	OCTN1	OCPOL1	OCTP0	OCTN0	OCPOL0	OCMODE1	OCMODE0
Test Register (note 1)	9	reserved	reserved	Map Internal Register	Connect RX Buffer 0 CPU	Connect TX Buffer CPU	Access Internal Bus	Normal RAM Connect	Float Output Driver
<b>Transmit Buffer</b>									
Identifier	10	ID.10	ID.9	ID.8	ID.7	ID.6	ID.5	ID.4	ID.3
RTR, Data Length Code	11	ID.2	ID.1	ID.0	RTR	DLC.3	DLC.2	DLC.1	DLC.0
bytes 1-8	12-19	Data	Data	Data	Data	Data	Data	Data	Data
<b>Receive Buffer 0/1</b>									
Identifier	20	ID.10	ID.9	ID.8	ID.7	ID.6	ID.5	ID.4	ID.3
RTR, Data Length Code	21	ID.2	ID.1	ID.0	RTR	DLC.3	DLC.2	DLC.1	DLC.0
bytes 1-8	22-29	Data	Data	Data	Data	Data	Data	Data	Data
Clock Divider	31	reserved	reserved	reserved	reserved	reserved	CD.2	CD.1	CD.0

**Notes**

1. The Test Register is used for production testing only.
2. Register 30 is not implemented.

## Stand-alone CAN-controller

## 82C200

## 6.2.1 CONTROL REGISTER (CR)

The contents of the Control Register are used to change the behaviour of the PCX82C200. Control bits may be set or reset by the attached microcontroller which uses the Control Register as a read/write memory.

Table 2 Description of the Control Register bits

CR	ADDRESS 0			
BIT	SYMBOL	NAME	VALUE	FUNCTION
CR.7	TM	Test Mode (note 1)	HIGH (enabled)	PCX82C200 enters Test Mode (normal operation impossible).
			LOW (disabled)	Normal operating mode.
CR.6	S	Sync (note 2)	HIGH (2 edges)	Bus-line transitions from recessive-to-dominant and vice versa are used for resynchronization (see sections 7.2 and 8).
			LOW (1 edge)	Only transitions from recessive-to-dominant are used for resynchronization.
CR.5	—	—	—	Reserved.
CR.4	OIE	Overrun Interrupt Enable	HIGH (enabled)	If the Data Overrun bit is set (see section 6.2.3), the microcontroller receives an Overrun Interrupt signal.
			LOW (disabled)	Microcontroller receives no Overrun Interrupt signal from the PCX82C200.
CR.3	EIE	Error Interrupt Enable	HIGH (enabled)	If the Error or Bus Status change (see section 6.2.3), the microcontroller receives an Error Interrupt signal.
			LOW (disabled)	Microcontroller receives no Error Interrupt signal.
CR.2	TIE	Transmit Interrupt Enable	HIGH (enabled)	When a message has been successfully transmitted or the transmit buffer is accessible again, (e.g. after an Abort Transmission command) the PCX82C200 transmits a Transmit Interrupt signal to the microcontroller.
			LOW (disabled)	No transmission of the Transmit Interrupt signal by the PCX82C200 to the microcontroller.
CR.1	RIE	Receive Interrupt Enable	HIGH (enabled)	When a message has been received without errors, the PCX82C200 transmits a Receive Interrupt signal to the microcontroller.
			LOW (disabled)	No transmission of the Receive Interrupt signal by the PCX82C200 to the microcontroller.
CR.0	RR	Reset Request (note 3)	HIGH (present)	Detection of a Reset Request results in the PCX82C200 aborting the current transmission or reception of a message and entering the reset state.
			LOW (absent)	On the HIGH-to-LOW transition of the Reset Request bit, the PCX82C200 returns to its normal operating state.

## Notes

1. The Test Mode is intended for factory testing and not for customer use.
2. The Sync bit should only be modified if the Reset Request bit is set HIGH (present), otherwise it is ignored. It is possible to set the Sync bit while the Reset Request bit is changed from HIGH to LOW.
3. During an external reset ( $\overline{RST} = \text{LOW}$ ) or when the Bus Status bit is set HIGH (Bus-Off), the IML forces the Reset Request HIGH (present). During an external reset the microcontroller cannot set the Reset Request bit LOW (absent). Therefore, after having set the Reset Request bit LOW (absent), the microcontroller must check this bit to ensure that the external reset pin is not being held HIGH (present). After the Reset Request bit is set LOW (absent) the PCX82C200 will wait for:
  - one occurrence of the Bus-Free signal (11 recessive bits, see section 8.9.6), if the preceding reset (Reset Request = HIGH) was due to an external reset or a microcontroller initiated reset
  - 128 occurrences of Bus-Free, if the preceding reset (Reset Request = HIGH) was due to a PCX82C200 initiated Bus-Off, before re-entering the Bus-On mode (see section 8.9).
 When Reset Request is set HIGH (present), for whatever reason, the control, command, status and interrupt bits are affected, see Table 3. When Reset Request is set HIGH (present) the registers at addresses 4 to 8 are accessible but the TBF is not.



## Stand-alone CAN-controller

## 82C200 CAN-controller

Table 3 Effects of setting the Reset Request bit HIGH (present)

TYPE	BIT	FUNCTION	EFFECT
Control	CR.7	Test Mode	LOW (disabled)
Command	CMR.4	Goto Sleep	LOW (wake-up)
	CMR.3	Clear Overrun Status	HIGH (clear)
	CMR.2	Release Receive Buffer	HIGH (released)
	CMR.1	Abort Transmission	LOW (absent)
	CMR.0	Transmission Request	LOW (absent)
Status	SR.7	Bus Status	LOW (Bus-On) (note 1)
	SR.6	Error Status	LOW (no error) (note 1)
	SR.5	Transmit Status	LOW (idle)
	SR.4	Receive Status	LOW (idle)
	SR.3	Transmission Complete Status	HIGH (complete)
	SR.2	Transmit Buffer Access	HIGH (released)
	SR.1	Data Overrun	LOW (absent)
	SR.0	Receive Buffer Status	LOW (empty)
Interrupt	IR.3	Overrun Interrupt	LOW (reset)
	IR.1	Transmit Interrupt	LOW (reset)
	IR.0	Receive Interrupt	LOW (reset)

## Note

1. Only after an external reset; see note 1 to Table 5 "Description of the Status Register bits".

## 6.2.2 COMMAND REGISTER (CMR)

A command bit initiates an action within the transfer layer of the PCX82C200. The Command Register appears to the microcontroller as a write only memory. If a read access is performed to this address the byte 11111111 (binary) is returned.

Table 4 Description of the Command Register bits

CMR	ADDRESS 1			
BIT	SYMBOL	NAME	VALUE	FUNCTION
CMR.7	—	—	—	Reserved.
CMR.6	—	—	—	Reserved.
CMR.5	—	—	—	Reserved.
CMR.4	GTS	GoTo Sleep (note 1)	HIGH (sleep)  LOW (wake up)	The PCX82C200 enters sleep mode, if the $\overline{\text{INT}} = \text{HIGH}$ (no interrupt signal from the PCX82C200 to the microcontroller pending or external source pending) and there is no bus activity. The PCX82C200 functions normally.
CMR.3	COS	Clear Overrun Status (note 2)	HIGH (clear)  LOW (no action)	The Data Overrun status bit is set to LOW (see section 6.2.3). No action.
CMR.2	RRB	Release Receive Buffer (note 3)	HIGH (released)  LOW (no action)	The receive buffer attached to the microcontroller is released. No action.
CMR.1	AT	Abort Transmission (note 4)	HIGH (present)  LOW (absent)	If not already in progress, a pending Transmission Request is cancelled. No action.
CMR.0	TR	transmission Request (note 5)	HIGH (present)  LOW (absent)	A message shall be transmitted. No action.

## Notes

1. The PCX82C200 will enter sleep mode, if GoTo Sleep is set HIGH (sleep), there is no bus activity and  $\overline{\text{INT}} = \text{HIGH}$  (inactive). After sleep mode is set, the CLK OUT signal continues until at least 15 bit times have passed. The PCX82C200 will wake up when one of the three previously mentioned conditions is negated: after GoTo Sleep is set LOW (wake up), there is bus activity or  $\overline{\text{INT}}$  is driven LOW (active). On wake up, the oscillator is started and a Wake-Up Interrupt (see section 6.2.4) is generated. A PCX82C200 which is sleeping and then awakened by bus activity will not be able to receive this message until it detects a Bus-Free signal (see section 8.9.6).
2. This command bit is used to acknowledge the Data Overrun condition signalled by the Data Overrun status bit. It may be given or set at the same time as a Release Receive Buffer command bit.
3. After reading the contents of the Receive Buffer (RBF0 or RBF1) the microcontroller must release this buffer by setting the Release Receive Buffer bit HIGH (released). This may result in another message becoming immediately available.
4. The Abort Transmission bit is used when the microcontroller requires the suspension of the previously requested transmission, for example to transmit an urgent message. A transmission already in progress is not stopped. In order to determine if the original message had been transmitted successfully, or aborted, the Transmission Complete status bit should be checked after the Transmit Buffer Access bit has been set HIGH (released) or a Transmit Interrupt has been generated (see section 6.2.4).
5. If the Transmission Request bit was set HIGH in a previous command, it cannot be cancelled by setting the Transmission Request bit LOW (absent). Cancellation of the requested transmission may be performed by setting the Abort Transmission bit HIGH (present).

## Stand-alone CAN-controller

## 82C200

## 6.2.3 STATUS REGISTER (SR)

The contents of the Status Register reflect the status of the PCX82C200 bus controller. The Status Register appears to the microcontroller as a read only memory.

Table 5 Description of the Status Register bits

SR	ADDRESS 2			
BIT	SYMBOL	NAME	VALUE	FUNCTION
SR.7	BS	Bus Status (note 1)	HIGH (Bus-Off)	The PCX82C200 is not involved in bus activities.
			LOW (Bus-On)	The PCX82C200 is involved in bus activities.
SR.6	ES	Error Status	HIGH (error)	At least one of the Error Counters (see section 8.10.3) has reached the microcontroller Warning Limit.
			LOW (ok)	Both Error Counters have not reached the Warning Limit.
SR.5	TS	Transmit Status (note 2)	HIGH (transmit)	The PCX82C200 is transmitting a message.
			LOW (idle)	No message is transmitted.
SR.4	RS	Receive Status (note 2)	HIGH (receive)	The PCX82C200 is receiving a message.
			LOW (idle)	No message is received.
SR.3	TCS	Transmission Complete Status (note 3)	HIGH (complete)	Last requested transmission has been successfully completed.
			LOW (incomplete)	Previously requested transmission is not yet completed.
SR.2	TBS	Transmit Buffer Access (note 3)	HIGH (released)	The microcontroller may write a message into the TBF.
			LOW (locked)	The microcontroller cannot access the Transmit Buffer. A message is either waiting for transmission or is in the process of being transmitted.
SR.1	DO	Data Overrun (note 4)	HIGH (overrun)	This bit is set HIGH (Overrun), when both Receive Buffers are full and the first byte of another message should be stored.
			LOW (absent)	No data overrun has occurred since the Clear Overrun command was given.
SR.0	RBS	Receive Buffer Status (note 5)	HIGH (full)	This bit is set when a new message is available.
			LOW (empty)	No message has become available since the last Release Receive Buffer command bit was set.

## Notes

- When the Bus Status bit is set HIGH (Bus-Off), the PCX82C200 will set the Reset Request bit HIGH (present). It will stay in this state until the microcontroller sets the Reset Request bit LOW (absent). Once this is completed the PCX82C200 will wait the minimum protocol-defined time (128 occurrences of the Bus-Free signal) before setting the Bus Status bit LOW (Bus-On), the Error Status bit LOW (ok) and resetting the Error Counters.
- If both the Receive Status and Transmit Status bits are LOW (idle) the CAN-bus is idle.
- If the microcontroller tries to write to the Transmit Buffer when the Transmit Buffer Access bit is LOW (locked), the written bytes will not be accepted and will be lost without this being signalled. The Transmission Complete Status bit is set LOW (incomplete) whenever the Transmission Request bit is set HIGH (present). If an Abort Transmission command is issued, the Transmit Buffer will be released. If the message, which was requested and then aborted, was not transmitted, the Transmission Complete Status bit will remain LOW.
- If Data Overrun = HIGH (Overrun) is detected, the currently received message is dropped. A transmitted message, granted acceptance, is also stored in a Receive Buffer. This occurs because it is not known if the PCX82C200 will lose arbitration and so become a receiver of the message. If no Receive Buffer is available, Data Overrun is signalled.
- If the command bit Release Receive Buffer is set HIGH (released) by the microcontroller, the Receive Buffer Status bit is set LOW (empty) by IML. When a new message is stored in any of the receive buffers, the Receive Buffer Status bit is set HIGH (full) again.

## Stand-alone CAN-controller

82C200

### 6.2.4 INTERRUPT REGISTER (IR)

The Interrupt Register allows the identification of an interrupt source. When one or more bits of this register are set, the INT pin is activated. All bits are reset by the PCX82C200 after this register is read by the microcontroller. This register appears to the microcontroller as a read only memory.

Table 6 Description of the Interrupt Register bits

IR		ADDRESS 3		
BIT	SYMBOL	NAME	VALUE	FUNCTION
IR.7	—	—	—	Reserved.
IR.6	—	—	—	Reserved.
IR.5	—	—	—	Reserved.
IR.4	WUI	Wake-Up Interrupt	HIGH (set) LOW (reset)	The Wake-Up Interrupt bit is set HIGH, when the sleep mode is left (see section 6.2.2). Wake-Up Interrupt bit is reset by a read access of Interrupt Register by the microcontroller.
IR.3	OI	Overrun Interrupt (note 1)	HIGH (set) LOW (reset)	This bit is set HIGH, if both Receive Buffers contain a message and the first byte of another message should be stored (passed acceptance), and the Overrun Interrupt Enable is HIGH (enabled). Overrun Interrupt bit is reset by a read access of Interrupt Register by the microcontroller.
IR.2	EI	Error Interrupt	HIGH (set) LOW (reset)	This bit is set on a change of either the Error Status or Bus Status bits (see section 6.2.3) if the Error Interrupt Enable is HIGH (enabled). The Error Interrupt bit is reset by a read access of the Interrupt Register by the microcontroller.
IR.1	TI	Transmit Interrupt	HIGH (set) LOW (reset)	This bit is set on a change of the Transmit Buffer Access bit from LOW to HIGH (released) and Transmit Interrupt Enable is HIGH (enabled). Transmit Interrupt bit will be reset after a read access of the Interrupt Register by the microcontroller.
IR.0	RI	Receive Interrupt (note 2)	HIGH (set) LOW (reset)	This bit is set when a new message is available in the Receive Buffer and the Receive Interrupt Enable bit is HIGH (enabled). Receive Interrupt bit is automatically reset by a read access of Interrupt Register by the microcontroller.

#### Notes

1. Overrun Interrupt bit (if enabled) and Data Overrun bit (see section 6.2.3) are set at the same time.
2. Receive Interrupt bit (if enabled) and Receive Buffer Status bit (see section 6.2.3) are set at the same time.

Stand-alone CAN-controller

82C200

6.2.5 ACCEPTANCE CODE REGISTER (ACR)

The Acceptance Code Register is part of the acceptance filter of the PCX82C200. This register can be accessed (read/write), if the Reset Request bit is set HIGH (present). When a message is received which passes the acceptance test and if there is an empty Receive Buffer, then the respective Descriptor and Data Field (see Fig.4) are sequentially stored in this empty buffer. In the case that there is no empty Receive Buffer, the Data Overrun bit is set HIGH (overrun), see sections 6.2.3 and 6.2.4. When the complete message has been correctly received the following occurs:

- the Receive Buffer Status bit is set HIGH (full)
- if the Receive Interrupt Enable bit is set HIGH (enabled), the Receive Interrupt is set HIGH (set).

The Acceptance Code bits (AC.7-AC.0) and the eight most significant bits of the message's Identifier (ID.10-ID.3) must be equal to those bit positions which are marked relevant by the Acceptance Mask bits (AM.7-AM.0). If the following equation is satisfied, acceptance is given:

$$[(ID.10 \dots ID.3) = (AC.7 \dots AC.0)] \text{ or } (AM.7 \dots AM.0) = 1111 \text{ 1111 binary}$$

During transmission of a message which passes the acceptance test, the message is also written to its own Receive Buffer. If no Receiver Buffer is available, Data Overrun is signalled because it is not known at the start of a message whether the PCX82C200 will lose arbitration and so become a receiver of the message.

Table 7 Acceptance Code Register bits

ACR		ADDRESS 4					
7	6	5	4	3	2	1	0
AC.7	AC.6	AC.5	AC.4	AC.3	AC.2	AC.1	AC.0

6.2.6 ACCEPTANCE MASK REGISTER (AMR)

The Acceptance Mask Register is part of the acceptance filter of the PCX82C200. This register can be accessed (read/write) if the Reset Request bit is set HIGH (present). The Acceptance Mask Register qualifies which of the corresponding bits of the acceptance code are "relevant" or "don't care" for acceptance filtering.

Table 8 Acceptance Mask Register bits

AMR		ADDRESS 5					
7	6	5	4	3	2	1	0
AM.7	AM.6	AM.5	AM.4	AM.3	AM.2	AM.1	AM.0

Table 9 Description of the Acceptance Mask Register bits

ACCEPTANCE MASK BIT	VALUE	COMMENTS
AM.7 to AM.0	HIGH (don't care)	This bit position is "don't care" for the acceptance of a message.
	LOW (relevant)	This bit position is "relevant" for acceptance filtering.



Stand-alone CAN-controller

82C200

6.2.7 Bus Timing Register 0 (BTR0)

The contents of Bus Timing Register 0 defines the values of Baud Rate Prescaler (BRP) and the Synchronization Jump Width (SJW). This register can be accessed (read/write) if the Reset Request bit is set HIGH (present).

Table 10 Bus Timing Register 0 bits

BTR0		ADDRESS 6					
7	6	5	4	3	2	1	0
SJW.1	SJW.0	BRP.5	BRP.4	BRP.3	BRP.2	BRP.1	BRP.0

Baud Rate Prescaler (BRP)

The period of the system clock  $t_{\text{SCL}}$  is programmable and determines the individual bit timing. The system clock is calculated using the following equation:

$t_{\text{SCL}} = 2t_{\text{CLK}} (32\text{BRP.5} + 16\text{BRP.4} + 8\text{BRP.3} + 4\text{BRP.2} + 2\text{BRP.1} + \text{BRP.0} + 1)$

$t_{\text{CLK}}$  = time period of the PCX82C200 oscillator.

Synchronization Jump Width (SJW)

To compensate for phase shifts between clock oscillators of different bus controllers, any bus controller must resynchronize on any relevant signal edge of the current transmission. The synchronization jump width defines the maximum number of clock cycles a bit period may be shortened or lengthened by one resynchronization:

$t_{\text{SJW}} = t_{\text{SCL}} (2\text{SJW.1} + \text{SJW.0} + 1)$

For further information on bus timing see sections 6.2.8 and 7.

6.2.8 Bus Timing Register 1 (BTR1)

The contents of Bus Timing Register 1 defines the length of the bit period, the location of the sample point and the number of samples to be taken at each sample point. This register may be accessed (read/write) if the Reset Request bit is set HIGH (present).

Table 11 Bus Timing Register 1 bits

BTR1		ADDRESS 7					
7	6	5	4	3	2	1	0
SAM	TSEG2.2	TSEG2.1	TSEG2.0	TSEG1.3	TSEG1.2	TSEG1.1	TSEG1.0

Sampling (SAM)

Table 12 Selection of sampling

BIT	VALUE	COMMENTS
SAM	HIGH (3 samples)	Three samples are taken.
	LOW (1 sample)	The bus is sampled once.

SAM = LOW (logic 0) is recommended for high speed buses (SAE class C), while SAM = HIGH (logic 1) is recommended for slow/medium speed buses (class A and B) where filtering of spikes on the bus-line is beneficial (see section 7.1.6).

Stand-alone CAN-controller

82C200

Time Segment 1 (TSEG1) and Time Segment 2 (TSEG2)

TSEG1 and TSEG2 determine the number of clock cycles per bit period and the location of the sample point:

tTSEG1 = tSCL (8TSEG1.3 + 4TSEG1.2 + 2TSEG1.1 + TSEG1.0 + 1)

tTSEG2 = tSCL (4TSEG2.2 + 2TSEG2.1 + TSEG2.0 + 1)

For further information on bus timing see sections 6.2.7 and 7.

6.2.9 OUTPUT CONTROL REGISTER (OCR)

The Output Control Register allows, under software control, the set-up of different output driver configurations. This register may be accessed (read/write) if the Reset Request bit is set HIGH (present).

Table 13 Output Control Register bits

OCR		ADDRESS 8					
7	6	5	4	3	2	1	0
OCTP1	OCTN1	OCPOL1	OCTP0	OCTN0	OCPOL0	OCMODE1	OCMODE0

If the PCX82C200 is in the sleep mode (Goto Sleep = HIGH) a recessive level is output on the TX0 and TX1 pins. If the PCX82C200 is in the reset state (Reset Request = HIGH) the output drivers are floating.

Normal Output Mode

In Normal Output Mode the bit sequence (TXD) is sent via TX0 and TX1. The voltage levels on the output driver pins TX1 and TX0 depend on both the driver characteristic programmed by OCTPx, OCTNx (float, pull-up, pull-down, push-pull) and the output polarity programmed by OCPOLx (see Fig.4).

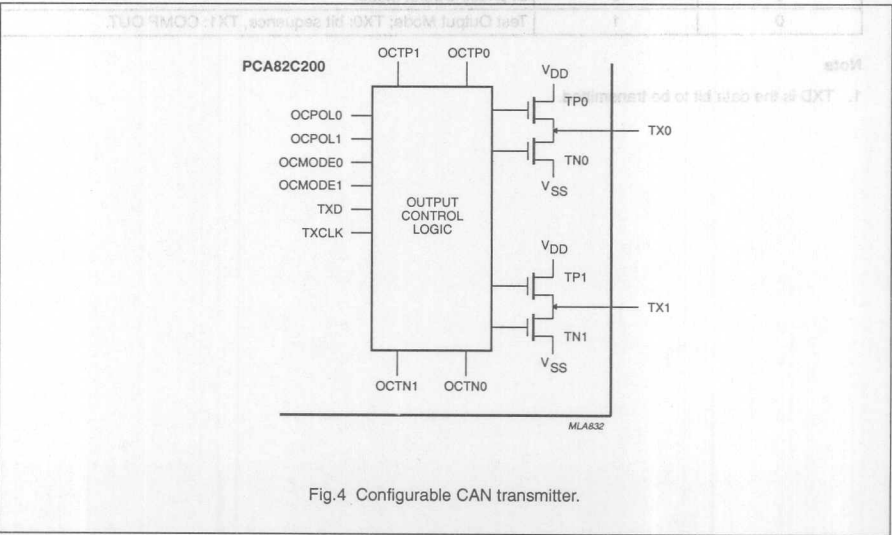


Fig.4 Configurable CAN transmitter.

## Stand-alone CAN-controller

## 82C200

### Clock Output Mode

For the TX0 pin this is the same as in Normal Output Mode. However, the data stream to TX1 is replaced by the transmit clock (TXCLK). The rising edge of the transmit clock (non inverted) marks the beginning of a bit period. The clock pulse width is  $t_{SCI}$ .

### Bi-phase Output Mode

In contrast to Normal Output Mode the bit representation is time variant and toggled. If the bus controllers are galvanically decoupled from the bus-line by a transformer, the bit stream is not allowed to contain a DC component. This is achieved by the following scheme. During recessive bits all outputs are deactivated (3-state). Dominant bits are sent alternately on TX0 and TX1, i.e. the first dominant bit is sent on TX0, the second is sent on TX1, and the third one is sent on TX0 again, etc.

### Test Output Mode

For the TX0 pin this is the same as in Normal Output Mode. To measure the delay time of the transmitter and receiver this mode connects the output of the input comparator (COMP OUT) with the input of the output driver TX1. This mode is used for production testing only.

The following two tables, Table 14 and Table 15, show the relationship between the bits of the Output Control Register and the two serial output pins TX0 and TX1 of the PCX82C200, connected to the serial bus (see Fig. 1).

Table 14 Description of the Output Mode bits

OCMODE1	OCMODE0	DESCRIPTION
1	0	Normal Output Mode; TX0, TX1: bit sequence (TXD; note 1).
1	1	Clock Output Mode; TX0: bit sequence, TX1: bus clock (TXCLK).
0	0	Bi-phase Output Mode.
0	1	Test Output Mode; TX0: bit sequence, TX1: COMP OUT.

### Note

1. TXD is the data bit to be transmitted.

Stand-alone CAN-controller

82C200

Table 15 Output pin set-up

DRIVE	OCTPx	OCTNx	OCPOLx	TXD	TPx (note 1)	TNx (note 2)	TXx (note 3)
Float	0	0	0	0	OFF	OFF	float
	0	0	0	1	OFF	OFF	float
	0	0	1	0	OFF	OFF	float
	0	0	1	1	OFF	OFF	float
Pull-down	0	1	0	0	OFF	ON	LOW
	0	1	0	1	OFF	OFF	float
	0	1	1	0	OFF	OFF	float
	0	1	1	1	OFF	ON	LOW
Pull-up	1	0	0	0	OFF	OFF	float
	1	0	0	1	ON	OFF	HIGH
	1	0	1	0	ON	OFF	HIGH
	1	0	1	1	OFF	OFF	float
Push/Pull	1	1	0	0	OFF	ON	LOW
	1	1	0	1	ON	OFF	HIGH
	1	1	1	0	ON	OFF	HIGH
	1	1	1	1	OFF	ON	LOW

Notes

1. TPx is the on-chip output transistor x, connected to V<sub>DD</sub>; x = 0 or 1.
2. TNx is the on-chip output transistor x, connected to V<sub>SS</sub>; x = 0 or 1.
3. TXx is the serial output level on pin TX0 or TX1. It is required that the output level on the CAN-bus is dominant with TXD = 0 and recessive with TXD = 1 (see section 8.1.1).

6.2.10 TEST REGISTER (TR)

The Test Register is used for production testing only.

Stand-alone CAN-controller

82C200

6.3 Transmit Buffer layout

The global layout of the Transmit Buffer is shown in Fig.3. This buffer serves to store a message from the microcontroller to be transmitted by the PCX82C200. It is subdivided into Descriptor and Data Field. The Transmit Buffer can be written to and read from by the microcontroller (see note 3 to Table 2).

6.3.1 DESCRIPTOR

Table 16 Descriptor Byte 1 (DSCR1)

DSCR1		ADDRESS 10					
7	6	5	4	3	2	1	0
ID.10	ID.9	ID.8	ID.7	ID.6	ID.5	ID.4	ID.3

Table 17 Descriptor Byte 2 (DSCR2)

DSCR2		ADDRESS 11					
7	6	5	4	3	2	1	0
ID.2	ID.1	ID.0	RTR	DLC.3	DLC.2	DLC.1	DLC.0

Identifier (ID)

The Identifier consists of 11 bits (ID.10 to ID.0). ID.10 is the most significant bit, which is transmitted first on the bus during the arbitration process. The Identifier acts as the message's name, used in a receiver for acceptance filtering and also determines the bus access priority during the arbitration process. The lower the binary value of the Identifier the higher the priority. This is due to the larger number of leading dominant bits during arbitration (see section 8.7 "Bus organization").

Remote Transmission Request bit (RTR)

Table 18 Description of the RTR bit

BIT	VALUE	COMMENTS
RTR	HIGH (remote)	Remote Frame will be transmitted by the PCX82C200.
	LOW (data)	Data Frame will be transmitted by the PCX82C200.

Data Length Code (DLC)

The number of bytes (Data Byte Count) in the Data Field of a message is coded by the Data Length Code. At the start of a Remote Frame transmission the Data Length Code is not considered due to the RTR bit being HIGH (remote). This forces the number of transmitted/received data bytes to be 0. Nevertheless, the Data Length Code must be specified correctly to avoid bus errors, if two CAN-controllers start a Remote Frame transmission simultaneously.

The range of the Data Byte Count is 0 to 8 bytes and coded as follows:

$$\text{Data Byte Count} = 8\text{DLC.3} + 4\text{DLC.2} + 2\text{DLC.1} + \text{DLC.0}$$

For reasons of compatibility no Data Byte Counts other than 0 to 8 should be used.

6.3.2 DATA FIELD

The number of transferred data bytes is determined by the Data Length Code. The first bit transmitted is the most significant bit of data byte 1 at address 12.



# Stand-alone CAN-controller

82C200

## 3.4 Receive Buffer layout

The layout of the Receive Buffer and the individual bytes correspond to the definitions given for the Transmit Buffer layout, except that the addresses start at 20 instead of 10 (see Fig.3).

## 3.5 Clock Divider Register (CDR)

The Clock Divider Register controls the CLK OUT frequency for the microcontroller (see Fig.1). It can be written to or read by the microcontroller. The default state of the register is divide by 12 for Motorola mode and divide by 2 for Intel mode. Values from 0 to 7 may be written into this register and will result in the CLK OUT frequencies shown in Table 20.

Table 19 Clock Divider Register bits

CDR				ADDRESS 31			
7	6	5	4	3	2	1	0
—	—	—	—	—	CD.2	CD.1	CD.0

### Note

Bits CDR.7 to CDR.3 are reserved.

Table 20 CLK OUT frequency selection

CD.2	CD.1	CD.0	CLK OUT FREQUENCY
0	0	0	$f_{CLK}/2$
0	0	1	$f_{CLK}/4$
1	0	0	$f_{CLK}/6$
0	1	1	$f_{CLK}/8$
1	0	0	$f_{CLK}/10$
1	0	1	$f_{CLK}/12$
1	1	0	$f_{CLK}/14$
1	1	1	$f_{CLK}$

### Note

1.  $f_{CLK}$  is the frequency of the oscillator.

## Stand-alone CAN-controller

82C200

### 7 BUS TIMING/SYNCHRONIZATION

The Bus Timing Logic (BTL) monitors the serial bus-line via the on-chip input comparator and performs the following functions (see section 5):

- monitors the serial bus-line level
- adjusts the sample point, within a bit period (programmable)
- samples the bus-line level using majority logic (programmable, 1 or 3 samples)
- synchronization to the bit stream:
  - hard synchronization at the start of a message
  - resynchronization during transfer of a message.

The configuration of the BTL is performed during the initialization of the PCX82C200. The BTL uses the following three registers:

- Control register (Sync)
- Bus Timing Register 0
- Bus Timing Register 1.

#### 7.1 Bit timing

A bit period is built up from a number of system clock cycles ( $t_{scl}$ ), see section 6.2.7. One bit period is the result of the addition of the programmable segments TSEG1 and TSEG2 and the general segment SYNCSEG (see sections 6.2.7 to 6.2.8).

##### 7.1.1 SYNCHRONIZATION SEGMENT (SYNCSEG)

The incoming edge of a bit is expected during this state; this state corresponds to one system clock cycle ( $1 \times t_{scl}$ ).

##### 7.1.2 TIME SEGMENT 1 (TSEG1)

This segment determines the location of the sampling point within a bit period, which is at the end of TSEG1. TSEG1 is programmable from 1 to 16 system clock cycles (see section 6.2.8).

The correct location of the sample point is essential for the correct functioning of a transmission. The following points must be taken into consideration:

- a Start-Of-Frame (see section 8.2.1) causes all PCX82C200's to perform a 'hard synchronization' (see section 7.2.1) on the first recessive-to-dominant edge. During arbitration, however, several PCX82C200's may simultaneously transmit. Therefore it may require twice the sum of bus-line, input comparator and the output driver delay times until the bus is stable. This is the propagation delay time.
- to avoid sampling at an incorrect position, it is necessary to include an additional synchronization buffer on both sides of the sample point. The main reasons for incorrect sampling are:
  - incorrect synchronization due to spikes on the bus-line
  - slight variations in the oscillator frequency of each PCX82C200 in the network, which results in a phase error.

Time Segment 1 consists of the segment for compensation of propagation delays and the synchronization buffer directly before the sample point (see Fig.5).

##### 7.1.3 TIME SEGMENT 2 (TSEG2)

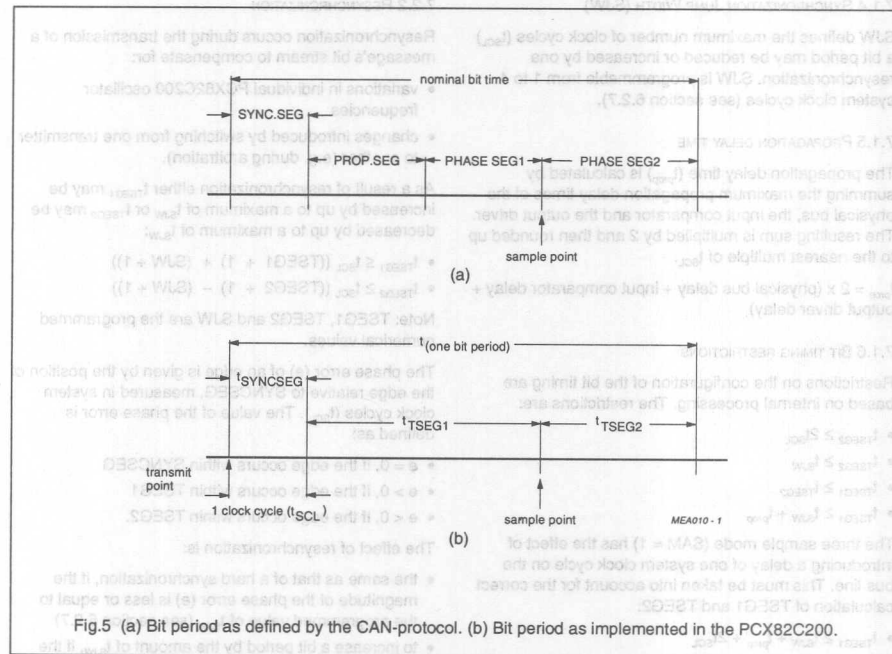
This time segment provides:

- additional time at the sample point for calculation of the subsequent bit levels (e.g. arbitration)
- synchronization buffer segment directly after the sample point (see section 7.1.2).

TSEG2 is programmable from 1 to 8 system clock cycles (see section 6.2.8).

## Stand-alone CAN-controller

## 82C200 CAN-controller



## Stand-alone CAN-controller

82C200

### 7.1.4 SYNCHRONIZATION JUMP WIDTH (SJW)

SJW defines the maximum number of clock cycles ( $t_{SCL}$ ) a bit period may be reduced or increased by one resynchronization. SJW is programmable from 1 to 4 system clock cycles (see section 6.2.7).

### 7.1.5 PROPAGATION DELAY TIME

The propagation delay time ( $t_{prop}$ ) is calculated by summing the maximum propagation delay times of the physical bus, the input comparator and the output driver. The resulting sum is multiplied by 2 and then rounded up to the nearest multiple of  $t_{SCL}$ .

$t_{prop} = 2 \times (\text{physical bus delay} + \text{input comparator delay} + \text{output driver delay})$

### 7.1.6 BIT TIMING RESTRICTIONS

Restrictions on the configuration of the bit timing are based on internal processing. The restrictions are:

- $t_{TSEG2} \geq 2t_{SCL}$
- $t_{TSEG2} \geq t_{SJW}$
- $t_{TSEG1} \geq t_{TSEG2}$
- $t_{TSEG1} \geq t_{SJW} + t_{prop}$

The three sample mode (SAM = 1) has the effect of introducing a delay of one system clock cycle on the bus-line. This must be taken into account for the correct calculation of TSEG1 and TSEG2:

- $t_{TSEG1} \geq t_{SJW} + t_{prop} + 2t_{SCL}$
- $t_{TSEG2} \geq 3t_{SCL}$

### 7.2 Synchronization

Synchronization is performed by a state machine which compares the incoming edge with its actual bit timing and adapts the bit timing by hard synchronization or resynchronization.

#### 7.2.1 HARD SYNCHRONIZATION

This type of synchronization occurs only at the beginning of a message. The PCX82C200 synchronizes on the first incoming recessive-to-dominant edge of a message (being the leading edge of a message's Start-Of-Frame bit; see section 7.1).

### 7.2.2 RESYNCHRONIZATION

Resynchronization occurs during the transmission of a message's bit stream to compensate for:

- variations in individual PCX82C200 oscillator frequencies
- changes introduced by switching from one transmitter to another (e.g. during arbitration).

As a result of resynchronization either  $t_{TSEG1}$  may be increased by up to a maximum of  $t_{SJW}$  or  $t_{TSEG2}$  may be decreased by up to a maximum of  $t_{SJW}$ :

- $t_{TSEG1} \leq t_{SCL} ((TSEG1 + 1) + (SJW + 1))$
- $t_{TSEG2} \geq t_{SCL} ((TSEG2 + 1) - (SJW + 1))$

Note: TSEG1, TSEG2 and SJW are the programmed numerical values.

The phase error ( $e$ ) of an edge is given by the position of the edge relative to SYNCSEG, measured in system clock cycles ( $t_{SCL}$ ). The value of the phase error is defined as:

- $e = 0$ , if the edge occurs within SYNCSEG
- $e > 0$ , if the edge occurs within TSEG1
- $e < 0$ , if the edge occurs within TSEG2.

The effect of resynchronization is:

- the same as that of a hard synchronization, if the magnitude of the phase error ( $e$ ) is less or equal to the programmed value of  $t_{SJW}$  (see section 6.2.7)
- to increase a bit period by the amount of  $t_{SJW}$ , if the phase error is positive and the magnitude of the phase error is larger than  $t_{SJW}$
- to decrease a bit period by the amount of  $t_{SJW}$ , if the phase error is negative and the magnitude of the phase error is larger than  $t_{SJW}$ .

#### 7.2.3 SYNCHRONIZATION RULES

The synchronization rules are as follows:

- only one synchronization within one bit time is used
- an edge is used for synchronization only if the value detected at the previous sample point differs from the bus value immediately after the edge
- hard synchronization is performed whenever there is a recessive-to-dominant edge during Bus-Idle (see section 7)

## Stand-alone CAN-controller

## 82C200

all other edges (recessive-to-dominant and optionally dominant-to-recessive edges if the Sync bit is set HIGH; see section 6.2.1) which are candidates for resynchronization will be used with the following exception:

- a transmitting PCX82C200 will not perform a resynchronization as a result of a recessive-to-dominant edge with positive phase error, if only these edges are used for resynchronization. This ensures that the delay times of the output driver and input comparator do not cause a permanent increase in the bit time.

## 8 COMMUNICATION PROTOCOL

## 8.1 Frame types

The PCX82C200 bus controller supports the four different CAN-protocol frame types for communication:

- Data Frame, to transfer data
- Remote Frame, request for data
- Error Frame, globally signal a (locally) detected error condition
- Overload Frame, to extend delay time of subsequent frames (an Overload Frame is not initiated by the PCX82C200).

## 8.1.1 BIT REPRESENTATION

There are two logical bit representations used in the CAN-protocol:

- a recessive bit on the bus-line appears only if all connected PCX82C200's send a recessive bit at that moment
- dominant bits always overwrite recessive bits i.e. the resulting bit level on the bus-line is dominant.

## 8.2 Data Frame

A Data Frame carries data from a transmitting PCX82C200 to one or more receiving PCX82C200's. A Data Frame is composed of seven different bit-fields:

- Start-Of-Frame
- Arbitration Field
- Control Field
- Data Field (may have a length of zero)
- CRC Field
- Acknowledge Field
- End-Of-Frame.

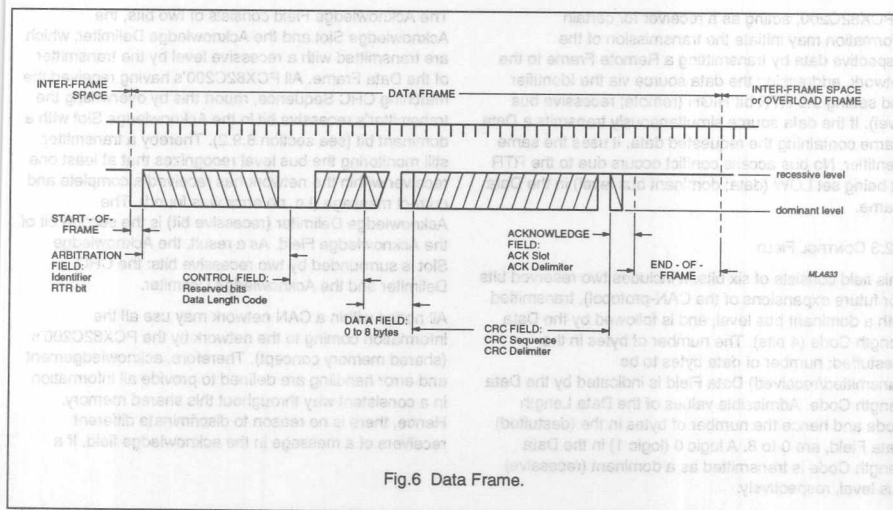


Fig.6 Data Frame.



## Stand-alone CAN-controller

## 82C200

### 8.2.1 START-OF-FRAME BIT

Signals the start of a Data Frame or Remote Frame. It consists of a single dominant bit used for hard synchronization of a PCX82C200 in receive mode.

### 8.2.2 ARBITRATION FIELD

Consists of the message Identifier and the RTR bit (see section 6.3.1). In the event of simultaneous message transmissions by two or more PCX82C200's the bus access conflict is solved by bit-wise arbitration, which is active during the transmission of the Arbitration Field.

#### Identifier

This 11-bit field is used to provide information about the message, as well as the bus access priority. It is transmitted in the order ID.10 to ID.0 (LSB). The situation that the seven most significant bits (ID.10 to ID.4) are all recessive must not occur.

An Identifier does not define which particular PCX82C200 will receive the frame, because a CAN based communication network does not discriminate between a point-to-point, multicast or broadcast communication.

#### Remote Transmission Request bit (RTR)

A PCX82C200, acting as a receiver for certain information may initiate the transmission of the respective data by transmitting a Remote Frame to the network, addressing the data source via the Identifier and setting the RTR bit HIGH (remote; recessive bus level). If the data source simultaneously transmits a Data Frame containing the requested data, it uses the same Identifier. No bus access conflict occurs due to the RTR bit being set LOW (data; dominant bus level) in the Data Frame.

### 8.2.3 CONTROL FIELD

This field consists of six bits. It includes two reserved bits (for future expansions of the CAN-protocol), transmitted with a dominant bus level, and is followed by the Data Length Code (4 bits). The number of bytes in the (destuffed; number of data bytes to be transmitted/received) Data Field is indicated by the Data Length Code. Admissible values of the Data Length Code and hence the number of bytes in the (destuffed) Data Field, are 0 to 8. A logic 0 (logic 1) in the Data Length Code is transmitted as a dominant (recessive) bus level, respectively.

### 8.2.4 DATA FIELD

The data, stored within the Data Field of the Transmit Buffer, are transmitted according to the Data Length Code. Conversely, data of a received Data Frame will be stored in the Data Field of a Receive Buffer. Data is stored byte-wise both for transmission by the microcontroller and on reception by the PCX82C200. The most significant bit of the first data byte (lowest address) is transmitted/received first.

### 8.2.5 CYCLIC REDUNDANCY CODE FIELD (CRC)

The CRC Field consists of the CRC Sequence (15 bits) and the CRC Delimiter (1 recessive bit). The Cyclic Redundancy Code (CRC) encloses the destuffed bit stream of the Start-Of-Frame, Arbitration Field, Control Field, Data Field and CRC Sequence. The most significant bit of the CRC Sequence is transmitted/received first. This frame check sequence, implemented in the PCX82C200, is derived from a cyclic redundancy code best suited for frames with a total bit count of less than 127 bits, see section 8.8.3 With Start-Of-Frame (dominant bit) included in the code word, any rotation of the code word can be detected by the absence of the CRC Delimiter (recessive bit).

### 8.2.6 ACKNOWLEDGE FIELD (ACK)

The Acknowledge Field consists of two bits, the Acknowledge Slot and the Acknowledge Delimiter, which are transmitted with a recessive level by the transmitter of the Data Frame. All PCX82C200's having received the matching CRC Sequence, report this by overwriting the transmitter's recessive bit in the Acknowledge Slot with a dominant bit (see section 8.9.2). Thereby a transmitter, still monitoring the bus level recognizes that at least one receiver within the network has received a complete and correct message (i.e. no error was found). The Acknowledge Delimiter (recessive bit) is the second bit of the Acknowledge Field. As a result, the Acknowledge Slot is surrounded by two recessive bits: the CRC Delimiter and the Acknowledge Delimiter.

All nodes within a CAN network may use all the information coming to the network by the PCX82C200's (shared memory concept). Therefore, acknowledgement and error handling are defined to provide all information in a consistent way throughout this shared memory. Hence, there is no reason to discriminate different receivers of a message in the acknowledge field. If a

## Stand-alone CAN-controller

## 82C200

node is disconnected from the network due to bus failure, this particular node is no longer part of the shared memory. To identify a 'lost node' additional and application specific precautions are required.

## 8.2.7 END-OF-FRAME

Each Data Frame or Remote Frame is delimited by the End-Of-Frame bit sequence which consists of seven recessive bits (exceeds the bit stuff width by two bits). Using this method a receiver detects the end of a frame independent of a previous transmission error because the receiver expects all bits up to the end of the CRC sequence to be coded by the method of bit-stuffing (see section 8.7.3). The bit-stuffing logic is deactivated during the End-Of-Frame sequence.

## 8.3 Remote Frame

A PCX82C200, acting as a receiver for certain information may initiate the transmission of the respective data by transmitting a Remote Frame to the network, addressing the data source via the Identifier and setting the RTR bit HIGH (remote; recessive bus level). The Remote Frame is similar to the Data Frame with the following exceptions:

- RTR bit is set HIGH
- Data Length Code is ignored
- no Data Field contained.

Note that the Data Length Code value should be the same as for the corresponding Data Frame (although this is ignored for a Remote Frame).

A Remote Frame is composed of six different bit fields:

- Start-Of-Frame
- Arbitration Field
- Control Field
- CRC-Field
- Acknowledge Field
- End-Of-Frame.

See section 8.2 for a more detailed explanation of the Remote Frame bit fields.

## 8.4 Error Frame

The Error Frame consists of two different fields. The first field is accomplished by the superimposing of Error Flags contributed from different PCX82C200s. The second field is the Error Delimiter (see Fig.7).

## 8.4.1 ERROR FLAG

There are two forms of an Error Flag:

- Active Error Flag, consists of six consecutive dominant bits
- Passive Error Flag, consists of six consecutive recessive bits unless it is overwritten by dominant bits from other PCX82C200's.

An error-active PCX82C200 (see section 8.9) detecting an error condition signals this by transmission of an Active Error Flag. This Error Flag's form violates the bit-stuffing law (see section 8.7.3) applied to all fields, from Start-Of-Frame to CRC Delimiter, or destroys the fixed form of the fields Acknowledge Field or End-Of-Frame (see Fig.6). Consequently, all other PCX82C200's detect an error condition and start transmission of an Error Flag. Therefore the sequence of dominant bits, which can be monitored on the bus, results from a superposition of different Error Flags transmitted by individual PCX82C200's. The total length of this sequence varies between six (minimum) and twelve (maximum) bits.

An error-passive PCX82C200 (see section 8.9) detecting an error condition tries to signal this by transmission of a Passive Error Flag. The error-passive PCX82C200 waits for six consecutive bits with identical polarity, beginning at the start of the Passive Error Flag. The Passive Error Flag is complete when these six identical bits have been detected.

## 8.4.2 ERROR DELIMITER

The Error Delimiter consists of eight recessive bits and has the same format as the Overload Delimiter. After transmission of an Error Flag, each PCX82C200 monitors the bus-line until it detects a transition from a dominant-to-recessive bit level. At this point in time, every PCX82C200 has finished sending its Error Flag and all PCX82C200's start transmission of seven recessive bits (plus the recessive bit at dominant-to-recessive transition, results in a total of eight recessive bits). After this event and an Intermission Field all error-active PCX82C200's within the network can start a transmission simultaneously.

If a detected error is signalled during transmission of a Data Frame or Remote Frame, the current message is spoiled and a retransmission of the message is initiated.

## Stand-alone CAN-controller

## PCX82C200 CAN-controller

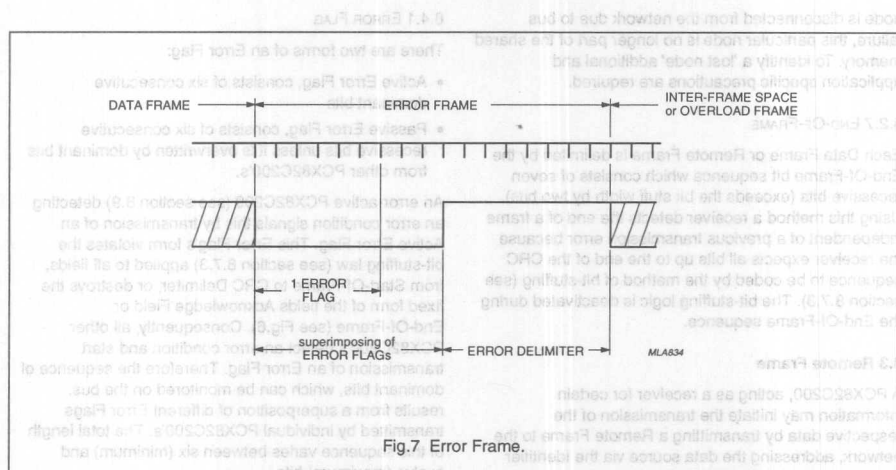


Fig.7 Error Frame.

If a PCX82C200 monitors any deviation of the Error Frame, a new Error Frame will be transmitted. Several consecutive Error Frame's may result in the PCX82C200 becoming error-passive and leaving the network unblocked.

In order to terminate an Error Flag correctly, an error-passive CAN-bus controller requires the bus to be Bus-Idle (see section 8.6.2) for at least three bit periods (if there is a local error at an error-passive receiver). Therefore a CAN-bus should not be 100% permanently loaded.

### 8.5 Overload Frame

The Overload Frame consists of two fields, the Overload Flag and the Overload Delimiter. There are two conditions in the CAN-protocol which lead to the transmission of an Overload Flag:

- condition 1; receiver circuitry requires more time to process the current data before receiving the next frame (receiver not ready)
- condition 2; detection of a dominant bit during Intermission Field (see section 8.6.1).

The transmission of an Overload Frame may only start:

- condition 1; during the first bit period of an expected Intermission Field

- condition 2; one bit period after detecting the dominant bit during Intermission Field.

The PCX82C200 will never initiate transmission of a condition 1 Overload Frame and will only react on a transmitted condition 2 Overload Frame, according to the CAN-protocol. No more than two Overload Frames are generated to delay a Data Frame or a Remote Frame. Although the overall form of the Overload Frame corresponds to that of the Error Frame, an Overload Frame does not initiate or require the retransmission of the preceding frame.

#### 8.5.1 OVERLOAD FLAG

The Overload Flag consists of six dominant bits and has a similar format to the Error Flag.

The Overload Flag's form corrupts the fixed form of the Intermission Field. All other PCX82C200's detecting the overload condition also transmit an Overload Flag (condition 2).

#### 8.5.2 OVERLOAD DELIMITER

The Overload Delimiter consists of eight recessive bits and takes the same form as the Error Delimiter. After transmission of an Overload Flag, each PCX82C200 monitors the bus-line until it detects a transition from a

## Stand-alone CAN-controller

## 82C200

dominant-to-recessive bit level. At this point in time, every PCX82C200 has finished sending its Overload Flag and all PCX82C200's start simultaneously transmitting seven more recessive bits.

### 8.6 Inter-Frame Space

Data Frames and Remote Frames are separated from preceding frames (all types) by an Inter-Frame Space, consisting of an Intermission Field and a Bus-Idle. Error-passive PCX82C200's also send a Suspend Transmission (see section 8.9.5) after transmission of a message. Overload Frames and Error Frames are not preceded by an Inter-Frame Space.

#### 8.6.1 INTERMISSION FIELD

The Intermission Field consists of three recessive bits. During an Intermission period, no frame transmissions will be started by any PCX82C200. An Intermission is required to have a fixed time period to allow a CAN-controller to execute internal processes prior to the next receive or transmit task.

#### 8.6.2 BUS-IDLE

The Bus-Idle time may be of arbitrary length (minimum 0 bit). The bus is recognized to be free and a CAN-controller having information to transmit may access the bus. The detection of a dominant bit level during Bus-Idle on the bus is interpreted as the Start-Of-Frame.

### 8.7 Bus organization

Bus organization is based on five basic rules described in the following paragraphs.

#### 8.7.1 BUS ACCESS

PCX82C200's only start transmission during the Bus-Idle state. All PCX82C200's synchronize on the leading edge of the Start-Of-Frame (hard synchronization).

#### 8.7.2 ARBITRATION

If two or more PCX82C200's simultaneously start transmitting, the bus access conflict is solved by a bit-wise arbitration process during transmission of the Arbitration Field.

During arbitration every transmitting PCX82C200 compares its transmitted bit level with the monitored bus level. Any PCX82C200 which transmits a recessive bit and monitors a dominant bus level immediately becomes

the receiver of the higher priority message on the bus without corrupting any information on the bus. Each message contains an unique Identifier and a RTR bit describing the type of data within the message.

The Identifier together with the RTR bit implicitly define the message's bus access priority. During arbitration the most significant bit of the Identifier is transmitted first and the RTR bit last. The message with the lowest binary value of the Identifier and RTR bit has the highest priority. A Data Frame has higher priority than a Remote Frame due to its RTR bit having a dominant level.

For every Data Frame there is an unique transmitter. For reasons of compatibility with other CAN-bus controllers, use of the Identifier binary bit pattern ID = 1111111XXXX (X being bits of arbitrary level) is forbidden. The number of available different Identifiers is  $2032 (2^{11} - 2^4)$ .

#### 8.7.3 CODING/DECODING

The following bit fields are coded using the bit-stuffing technique:

- Start-Of-Frame
- Arbitration Field
- Control Field
- Data Field
- CRC Sequence.

When a transmitting PCX82C200 detects five consecutive bits of identical polarity to be transmitted, a complementary (stuff) bit is inserted into the transmitted bit-stream.

When a receiving PCX82C200 has monitored five consecutive bits with identical polarity in the received bit streams of the above described bit fields, it automatically deletes the next received (stuff) bit. The level of the deleted stuff bit has to be the complement of the previous bits; otherwise a Stuff Error will be detected and signalled (see section 8.8.2).

The remaining bit fields or frames are of fixed form and are not coded or decoded by the method of bit-stuffing.

The bit-stream in a message is coded according to the Non-Return-to-Zero (NRZ) method, i.e. during a bit period, the bit level is held constant, either recessive or dominant.

## Stand-alone CAN-controller

## PCX82C200

### 8.7.4 ERROR SIGNALING

A PCX82C200 which detects an error condition, transmits an Error Flag. Whenever a Bit Error, Stuff Error, Form Error or an Acknowledgement Error is detected, transmission of an Error Flag is started at the next bit. Whenever a CRC Error is detected, transmission of an Error Flag starts at the bit following the Acknowledge Delimiter, unless an Error Flag for another error condition has already started. An Error Flag violates the bit-stuffing law or corrupts the fixed form bit fields. A violation of the bit-stuffing law affects any PCX82C200 which detects the error condition. These devices will also transmit an Error Flag.

An error-passive PCX82C200 (see section 8.9) which detects an error condition, transmits a Passive Error Flag. A Passive Error Flag is not able to interrupt a current message at PCX82C200's, but this type of Error Flag may be cancelled by other PCX82C200's. After having detected an error condition, an error-passive PCX82C200 will wait for six consecutive bits with identical polarity and when monitoring them, interpret them as an Error Flag.

After transmission of an Error Flag, each PCX82C200 monitors the bus-line until it detects a transition from a dominant-to-recessive bit level. At this point in time, every PCX82C200 has finished transmitting its Error Flag and all PCX82C200's start transmitting seven additional recessive bits (Error Delimiter, see section 8.4.2).

The message format of a Data Frame or Remote Frame is defined in such a way, that all detectable errors can be signalled within the message transmission time and therefore, it is very simple for a PCX82C200 to associate an Error Frame to the corresponding message and to initiate retransmission of the corrupted message.

If a PCX82C200 monitors any deviation of the fixed form of an Error Frame, it transmits a new Error Frame.

### 8.7.5 OVERLOAD SIGNALING

Some CAN-controllers (but not the PCX82C200) require to delay the transmission of the next Data Frame or Remote Frame by transmitting one or more Overload Frames. The transmission of an Overload Frame must start during the first bit of an expected Intermission. Transmission of Overload Frames which are reactions on a dominant bit during an expected Intermission Field, start one bit after this event.

Though the format of Overload Frame and Error Frame are identical, they are treated differently. Transmission of an Overload Frame during Intermission Field does not initiate the retransmission of any previous Data Frame or Remote Frame.

If a CAN-controller which transmitted an Overload Frame monitors any deviation of its fixed form, it transmits an Error Frame.

### 8.8 Error detection

The processes described in the following paragraphs are implemented in the PCX82C200 for error detection.

#### 8.8.1 BIT ERROR

A transmitting PCX82C200 monitors the bus on a bit-by-bit basis. If the bit level monitored is different from the transmitted one, a Bit Error is signalled. The exceptions being:

- during the Arbitration Field, a recessive bit can be overwritten by a dominant bit. In this case, the PCX82C200 interprets this as a loss of arbitration
- during the Acknowledge Slot, only the receiving PCX82C200's are able to recognize a Bit Error.

#### 8.8.2 STUFF ERROR

The following bit fields are coded using the bit-stuffing technique:

- Start-Of-Frame
- Arbitration Field
- Control Field
- Data Field
- CRC Sequence.

There are two possible ways of generating a Stuff Error:

- the disturbance generates more than the allowed five consecutive bits with identical polarity. These errors are detected by all PCX82C200's
- a disturbance falsifies one or more of the five bits preceding the stuff bit. This error situation is not recognized as a Stuff Error by the receivers. Therefore, other error detection processes may detect this error condition such as: CRC check, format violation at the receiving PCX82C200's or Bit Error detection by the transmitting PCX82C200.



## Stand-alone CAN-controller

## 82C200

## 8.8.3 CRC ERROR

To ensure the validity of a transmitted message all receivers perform a CRC check. Therefore, in addition to the (destuffed) information digits (Start-Of-Frame up to Data Field), every message includes some control digits (CRC Sequence; generated by the transmitting PCX82C200 of the respective message) used for error detection.

The code used for the PCX82C200 bus controller is a (shortened) BCH code, extended by a parity check and has the following attributes:

- 127 bits as maximum length of the code
- 112 bits as maximum number of information digits (maximum 83 bits are used by PCX82C200)
- length of the CRC Sequence amounts to 15 bits
- Hamming distance  $d = 6$ .

As a result, (d-1) random errors are detectable (some exceptions exist).

The CRC Sequence is calculated by the following procedure:

1. the destuffed bit stream consisting of Start-Of-Frame up to the Data Field (if present) is interpreted as a polynomial with coefficients of 0 or 1
2. this polynomial is divided (modulo-2) by the following generator polynomial:

$$f(X) = (X^{14} + X^9 + X^8 + X^6 + X^5 + X^4 + X^2 + X + 1) (X + 1) = 1100010110011001 \text{ binary.}$$

The remainder of this polynomial division is the CRC Sequence which includes a parity check. Burst errors are detected up to a length of 15 [degree of  $f(X)$ ]. Multiple errors (number of disturbed bits at least  $d = 6$ ) are not detected with a residual error probability of  $2^{-15}$  ( $\approx 3 \times 10^{-5}$ ) by CRC check only.

## 8.8.4 FORM ERROR

Form Errors result from violation of the fixed form of the following bit fields:

- End-Of-Frame
- Intermission
- Acknowledge Delimiter
- CRC Delimiter.

During the transmission of these bit fields an error condition is recognized if a dominant bit level instead of a recessive one is detected.

## 8.8.5 ACKNOWLEDGEMENT ERROR

This is detected by a transmitter whenever it does not monitor a dominant bit during the Acknowledge Slot.

## 8.8.6 ERROR DETECTION BY AN ERROR FLAG OF ANOTHER PCX82C200

The detection of an error is signalled by transmitting an Error Flag. An Active Error Flag causes a Stuff Error, a Bit Error or a Form Error at all other PCX82C200's.

## 8.8.7 ERROR DETECTION CAPABILITIES

Errors which occur at all PCX82C200's (global errors) are 100% detected. For local errors, i.e. for errors occurring at some PCX82C200's only, the shortened BCH code, extended by a parity check, has the following error detection capabilities:

- up to five single bit errors are 100% detected, even if they are distributed randomly within the code
- all single bit errors are detected if their total number (within the code) is odd
- the residual error probability of the CRC check amounts to  $3 \times 10^{-5}$ . As an error may be detected not only by CRC check but also by other detection processes described in sections 8.8.1 to 8.8.5, the residual error probability is several magnitudes less than  $3 \times 10^{-5}$  for undetected errors.

## 8.9 Error confinement (definitions)

## 8.9.1 Bus-Off

A PCX82C200 which has too many unsuccessful transmissions, relative to the number of successful transmissions, will enter the Bus-Off state. It remains in this state, neither receiving nor transmitting messages until the Reset Request bit is set LOW (absent) and both Error Counters are set to '0' (see note 1 to Table 5 and section 8.10.3).

## 8.9.2 ACKNOWLEDGE (ACK)

A PCX82C200 which has received a valid message correctly, indicates this to the transmitter by transmitting a dominant bit level on the bus during the Acknowledge Slot, independent of accepting or rejecting the message.

## Stand-alone CAN-controller

## 82C200

### 8.9.3 ERROR-ACTIVE

An error-active PCX82C200 is in its normal operating state able to receive and to transmit normally and also to transmit an active Error Flag (see section 8.10.3).

### 8.9.4 ERROR-PASSIVE

An error-passive PCX82C200 may transmit or receive messages normally. In the case of a detected error condition it transmits a Passive Error Flag, instead of an Active Error Flag. Hence the influence on bus activities by an error-passive PCX82C200 (e.g. due to a malfunction) is reduced.

### 8.9.5 SUSPEND TRANSMISSION

After an error-passive PCX82C200 has transmitted a message, it sends eight recessive bits after the Intermission Field and then checks for Bus-Idle. If during Suspend Transmission another PCX82C200 starts transmitting a message the suspended PCX82C200 will become the receiver of this message; otherwise being in Bus-Idle it may start to transmit a further message.

### 8.9.6 START-UP

A PCX82C200 which was either switched off or is in the Bus-Off state, must run a start-up routine in order to:

- synchronize with other available PCX82C200's, before starting to transmit. Synchronizing is achieved, when 11 recessive bits, equivalent to Acknowledge Delimiter, End-Of-Frame and Intermission Field, have been detected (Bus-Free)
- wait for other PCX82C200s without passing into the Bus-Off state (due to a missing acknowledge), if there is no other PCX82C200 currently available.

## 8.10 Aims of error confinement

### 8.10.1 DISTINCTION OF SHORT AND LONG-LASTING DISTURBANCES

The microcontroller must be informed when there are long-lasting disturbances and when bus activities have returned to normal operation. During long-lasting disturbances, a PCX82C200 enters the Bus-Off state and the microcontroller may use default values.

Minor disturbances of bus activities will not affect a PCX82C200. In particular, a PCX82C200 does not enter the Bus-Off state or inform the microcontroller of a short-lasting bus disturbance.

### 8.10.2 DETECTION AND LOCALIZATION OF HARDWARE DISTURBANCES AND DEFECTS

The rules for error confinement are defined by the CAN-protocol specification (and implemented in the PCX82C200), in that the PCX82C200, being nearest to the error-locus, reacts with a high probability, the quickest (i.e. becomes error-passive or Bus-Off), hence errors can be localized and their influence on normal bus activities is minimized.

### 8.10.3 ERROR CONFINEMENT

All PCX82C200's contain a Transmit Error Counter and a Receive Error Counter, which registers errors during the transmission and the reception of messages, respectively.

If a message is transmitted or received correctly, the count is decreased. In the event of an error, the count is increased. The Error Counters have a non-proportional method of counting: an error causes a larger counter increase than a correctly transmitted/received message causes the count to decrease. Over a period of time this may result in an increase in error counts, even if there are fewer corrupted messages than uncorrupted ones. The level of the Error Counters reflect the relative frequency of disturbances. The ratio of increase/decrease depends on the acceptable ratio of invalid/valid messages on the bus and is hardware implemented to eight.

If one of the Error Counters exceeds the Warning Limit of 96 error points, indicating a significant accumulation of error conditions, this is signalled by the PCX82C200 (Error Status, Error Interrupt).

A PCX82C200 operates in the error-active mode until it exceeds 127 error points on one of its Error Counters. At this point it will enter the error-passive state.

A transmit error which exceeds 255 error points results in the PCX82C200 entering the Bus-Off state.

Stand-alone CAN-controller

82C200

9 LIMITING VALUES

Limiting values in accordance with the Absolute Maximum Rating System (IEC134)

SYMBOL	PARAMETER	MIN.	MAX.	UNIT
$V_{DD}$	supply voltage range	4.5	5.5	V
$I_I$	input/output current on any pin except for TX0 and TX1	—	$\pm 10$	mA
$I_{OT}$	sink current of TX0 and TX1 together (note 1)	—	28	mA
$I_{OT}$	source current of TX0 and TX1 together (note 1)	—	—20	mA
$T_{amb}$	operating ambient temperature range: PCA82C200	—40	+125	$^{\circ}\text{C}$
	PCF82C200	—40	+85	$^{\circ}\text{C}$
$T_{stg}$	storage temperature range	—65	+150	$^{\circ}\text{C}$
$P_{tot}$	total power dissipation (note 2)	—	1	W

Notes

- $I_{OT}$  is allowed in case of a bus failure condition because then the TX-outputs are switched off automatically after a short time (Bus-Off state). During normal operation  $I_{OT}$  is a peak current, permitted for  $t < 100$  ms. The average output current must not exceed 10 mA for each TX-output.
- The value is based on the maximum allowable die temperature and the thermal resistance of the package, not on device power consumption.

10 DC CHARACTERISTICS

$V_{DD} \leq 5$  V  $\pm 10\%$ ;  $V_{SS} = 0$  V;  $T_{amb} = -40$  to  $+125$   $^{\circ}\text{C}$  for the PCA82C200 and  $T_{amb} = -40$  to  $+85$   $^{\circ}\text{C}$  for the PCF82C200. All voltages measured with respect to  $V_{SS}$  unless otherwise specified

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
<b>Supply</b>					
$V_{DD}$	supply voltage range		4.5	5.5	V
$I_{DD}$	supply current: operating	$\overline{RST} = V_{SS};$ $f_{CLK} = 16$ MHz (note 1)	—	15	mA
$I_{sm}$	sleep mode	oscillator inactive (note 2)	—	40	$\mu\text{A}$
<b>Inputs</b>					
$V_{IL1}$	LOW level input voltage (except XTAL1, RX0 and RX1)		—0.5	0.8	V
$V_{IL2}$	XTAL1 LOW level input voltage		—	$0.2V_{DD}$	V
$V_{IH1}$	HIGH level input voltage (except XTAL1, $\overline{RST}$ , RX0 and RX1)		3.2	$V_{DD} + 0.5$	V
$V_{IH2}$	XTAL1 HIGH level input voltage		$0.7V_{DD}$	—	V
$V_{IH3}$	$\overline{RST}$ HIGH level input voltage		3.3	$V_{DD} + 0.5$	V
$I_{LI}$	input leakage current (except XTAL1, RX0 and RX1)	$0.45$ V $< V_I < V_{DD}$	—	$\pm 10$	$\mu\text{A}$

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
<b>Outputs</b>					
$V_{OL}$	LOW level output voltage (except XTAL2, TX0 and TX1)	$I_{OL} = 1.6 \text{ mA}$	—	0.45	V
$V_{OH1}$	HIGH level output voltage (except TX0, TX1, INT and CLK OUT)	$I_{OH} = -80 \mu\text{A}$	2.4	—	V
$V_{OH2}$	CLK OUT HIGH level output voltage	$I_{OH} = -80 \mu\text{A}$	$0.8V_{DD}$	—	V
<b>CAN input comparator</b>					
		$V_{DD} = 5 \text{ V} \pm 5\%$ ; $1.4 \text{ V} < V_I < V_{DD} - 1.4 \text{ V}$			
$V_{DIF}$	differential input voltage	note 3	$\pm 42$		mV
$V_{HYST}$	hysteresis voltage	note 3	12	45	mV
$I_I$	input current		—	$\pm 400$	nA
<b>CAN output driver</b>					
		$V_{DD} = 5 \text{ V} \pm 5\%$			
$V_{OLT}$	TX0 and TX1 output voltage LOW	$I_O = 1.2 \text{ mA}$ (note 3)	—	0.1	V
		$I_O = 10 \text{ mA}$	—	1.0	V
$V_{OHT}$	TX0 and TX1 output voltage HIGH	$I_O = 1.2 \text{ mA}$ (note 3)	$V_{DD} - 0.1$	—	V
		$I_O = 10 \text{ mA}$	$V_{DD} - 1.0$	—	V

#### Notes

- (AD0 – AD7) = ALE =  $\overline{RD}$  =  $\overline{WR}$  =  $\overline{CS}$  =  $V_{DD}$ ; MODE =  $V_{SS}$ ; RX0 = 2.7 V; RX1 = 2.3 V; XTAL1 =  $0.5V_{DD} - 0.5 \text{ V}$ ; all outputs unloaded.
- (AD0 – AD7) = ALE =  $\overline{RD}$  =  $\overline{WR}$  = INT =  $\overline{RST}$  =  $\overline{CS}$  = MODE = RX0 =  $V_{DD}$ ; RX1 = XTAL1 =  $V_{SS}$ ; all outputs unloaded.
- Not tested during production.

# Stand-alone CAN-controller

82C200

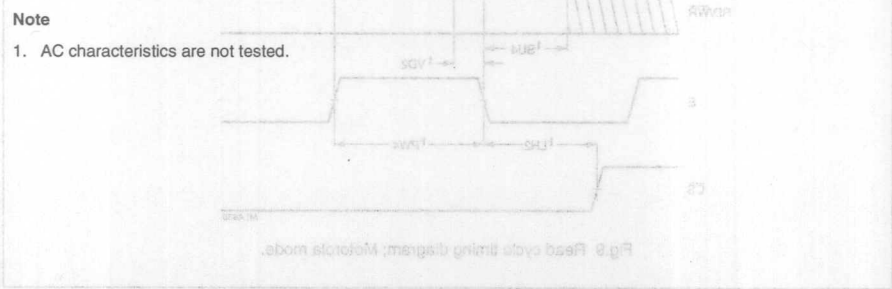
## 11 AC CHARACTERISTICS

$V_{DD} = 5\text{ V} \pm 10\%$ ;  $V_{SS} = 0\text{ V}$ ;  $C_L = 50\text{ pF}$  (output pins);  $T_{amb} = -40\text{ to }+85/125\text{ }^{\circ}\text{C}$ ; unless otherwise specified (note 1)

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
$f_{CLK}$	oscillator frequency		3	16	MHz
$t_{SU1}$	address set-up to ALE/AS LOW		10	—	ns
$t_{HD1}$	address hold time		22	—	ns
$t_{PW1}$	ALE/AS pulse width		35	—	ns
$t_{VD1}$	$\overline{RD}$ LOW to valid data output	Intel mode	—	60	ns
$t_{VD2}$	E HIGH to valid data output	Motorola mode	—	60	ns
$t_{bF1}$	data float after $\overline{RD}$ HIGH	Intel mode	10	55	ns
$t_{bF2}$	data float after E LOW	Motorola mode	10	55	ns
$t_{SU2}$	input data set-up to $\overline{WR}$ HIGH	Intel mode	30	—	ns
$t_{HD2}$	input data hold after $\overline{WR}$ HIGH	Intel mode	13	—	ns
$t_{HH1}$	$\overline{WR}$ HIGH to next ALE HIGH		23	—	ns
$t_{LH3}$	E LOW to next AS HIGH	Motorola mode	23	—	ns
$t_{SU3}$	input data set-up to E LOW	Motorola mode	30	—	ns
$t_{HD3}$	input data hold after E LOW	Motorola mode	25	—	ns
$t_{LL1}$	ALE LOW to $\overline{WR}$ LOW	Intel mode	10	—	ns
$t_{LL2}$	ALE LOW to $\overline{RD}$ LOW	Intel mode	10	—	ns
$t_{LH1}$	AS LOW to E HIGH	Motorola mode	10	—	ns
$t_{SU4}$	set-up time of $\overline{RD}/\overline{WR}$ to E HIGH	Motorola mode	20	—	ns
$t_{PW2}$	$\overline{WR}$ pulse width	Intel mode	170	—	ns
$t_{PW3}$	$\overline{RD}$ pulse width	Intel mode	170	—	ns
$t_{PW4}$	E pulse width	Motorola mode	170	—	ns
$t_{LL3}$	CS LOW to $\overline{WR}$ LOW	Intel mode	0	—	ns
$t_{LL4}$	CS LOW to $\overline{RD}$ LOW	Intel mode	0	—	ns
$t_{LH2}$	CS LOW to E HIGH	Motorola mode	0	—	ns
Input comparator/output driver					
$t_{sd}$	sum of the input and output delays	$V_{DD} = 5\text{ V} \pm 5\%$ ; $V_{DIF} = \pm 42\text{ mV}$ ; $1.4\text{ V} < V_I < V_{DD} - 1.4\text{ mV}$	—	62	ns

### Note

1. AC characteristics are not tested.





Stand-alone CAN-controller

82C200

11.1 AC timing diagrams

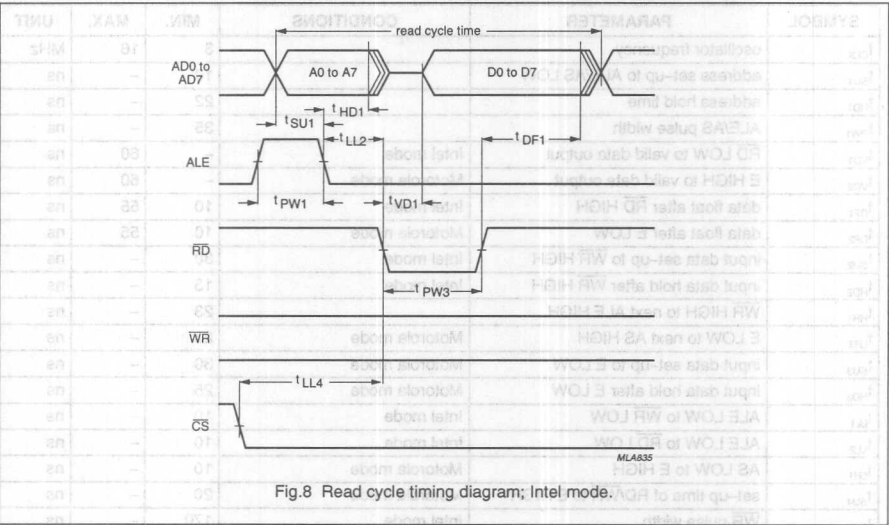


Fig.8 Read cycle timing diagram; Intel mode.

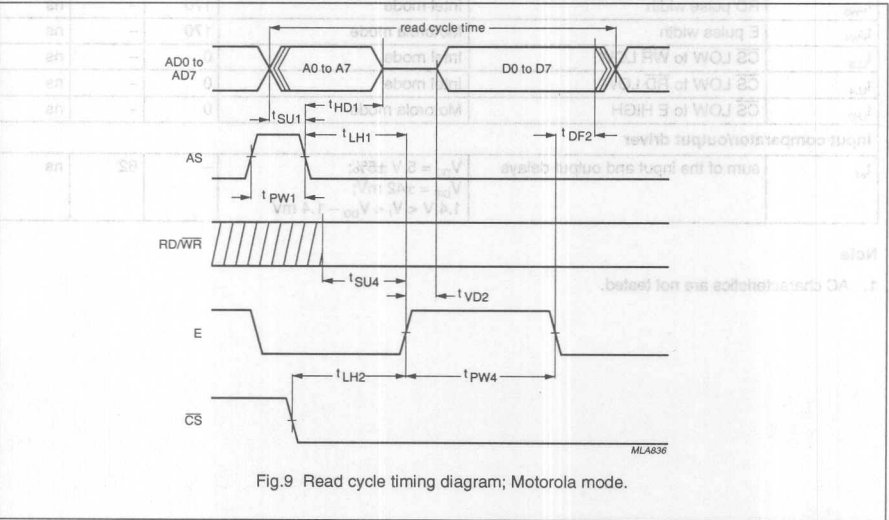
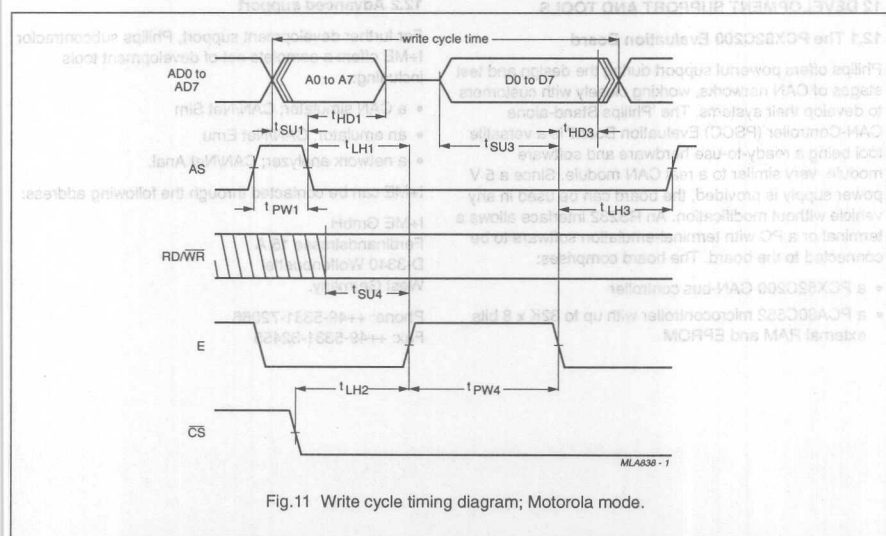
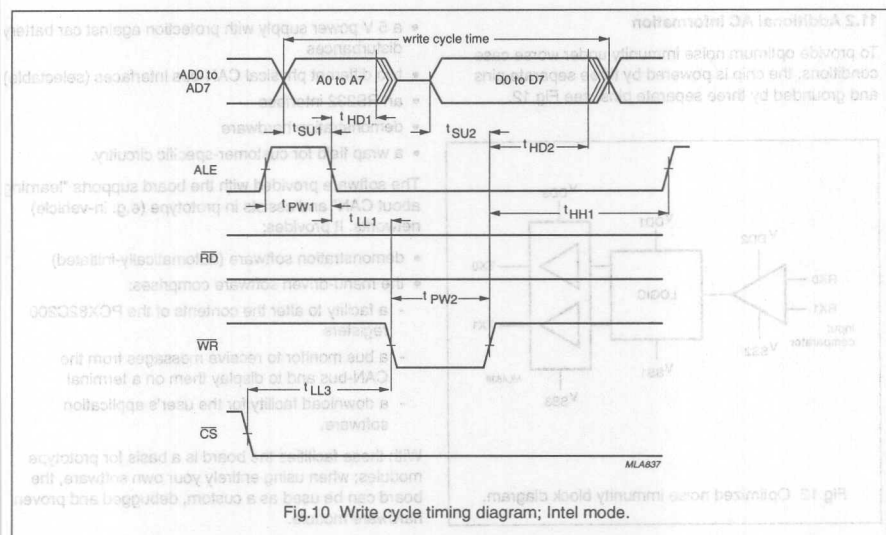


Fig.9 Read cycle timing diagram; Motorola mode.

## Stand-alone CAN-controller

82C200  
stand-alone CAN-controller

## Stand-alone CAN-controller

82C200

## 11.2 Additional AC information

To provide optimum noise immunity under worse case conditions, the chip is powered by three separate pins and grounded by three separate pins, see Fig.12.

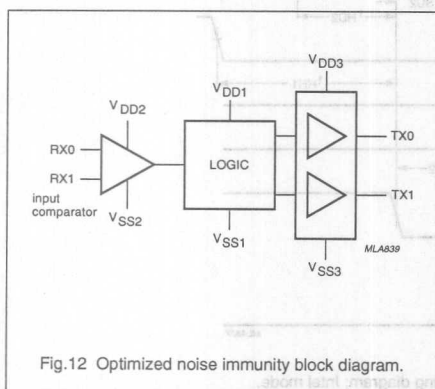


Fig.12 Optimized noise immunity block diagram.

- a 5 V power supply with protection against car battery disturbances
- two different physical CAN-bus interfaces (selectable)
- an RS232 interface
- demonstration hardware
- a wrap field for customer-specific circuitry.

The software provided with the board supports "learning about CAN" and assists in prototype (e.g. in-vehicle) networks. It provides:

- demonstration software (automatically-initiated)
- the menu-driven software comprises:
  - a facility to alter the contents of the PCX82C200 registers
  - a bus monitor to receive messages from the CAN-bus and to display them on a terminal
  - a download facility for the user's application software.

With these facilities the board is a basis for prototype modules; when using entirely your own software, the board can be used as a custom, debugged and proven hardware module.

## 12 DEVELOPMENT SUPPORT AND TOOLS

## 12.1 The PCX82C200 Evaluation Board

Philips offers powerful support during the design and test stages of CAN networks, working closely with customers to develop their systems. The 'Philips Stand-alone CAN-Controller (PSCC) Evaluation Board' is a versatile tool being a ready-to-use hardware and software module, very similar to a real CAN module. Since a 5 V power supply is provided, the board can be used in any vehicle without modification. An RS232 interface allows a terminal or a PC with terminal-emulation software to be connected to the board. The board comprises:

- a PCX82C200 CAN-bus controller
- a PCA80C552 microcontroller with up to 32K x 8 bits external RAM and EPROM

## 12.2 Advanced support

For further development support, Philips subcontractor I+ME offers a complete set of development tools including:

- a CAN simulator; CAN/Net Sim
- an emulator; CAN/Net Emu
- a network analyzer; CAN/Net Anal.

I+ME can be contacted through the following address:

I+ME GmbH  
Ferdinandstrasse 15 A  
D-3340 Wolfenbuettel  
West Germany.

Phone: ++49-5331-72066  
Fax: ++49-5331-32455

CAN controller interface

PCA82C250

FEATURES

- Fully compatible with the ISO/DIS 11898 standard
- High speed (up to 1 Mbaud)
- Bus lines protected against transients in an automotive environment
- Slope control to reduce radio frequency interference (RFI)
- Thermally protected
- Short-circuit proof to battery and ground
- Low current standby mode
- An unpowered node does not disturb the bus lines
- At least 110 nodes can be connected.

GENERAL DESCRIPTION

The PCA82C250 is the interface between the CAN protocol controller and the physical bus. It is primarily intended for high-speed applications (up to 1 Mbaud) in cars. The device provides differential transmit capability to the bus and differential receive capability to the CAN controller

QUICK REFERENCE DATA

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
V <sub>CC</sub>	supply voltage		4.5	—	5.5	V
I <sub>CC</sub>	supply current		—	—	170	μA
f <sub>bit</sub>	maximum transmission speed	NRZ	1	—	—	Mbaud
V <sub>CAN</sub>	CANH, CANL input/output voltage		−8	−2	+18	V
ΔV	differential bus voltage		1.5	—	3.0	V
t <sub>d</sub>	propagation delay	high-speed mode	—	—	50	ns
T <sub>amb</sub>	operating ambient temperature		−40	—	+125	°C

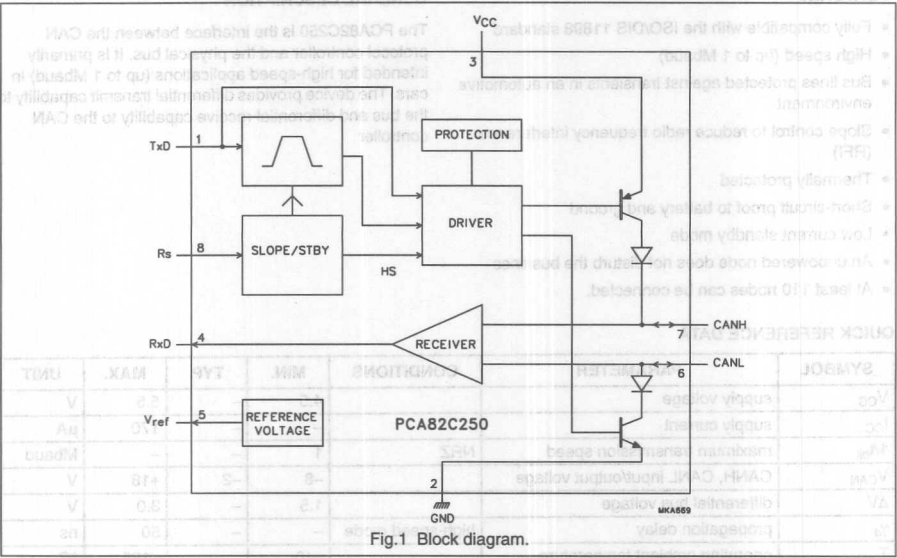
ORDERING INFORMATION

EXTENDED TYPE NUMBER	PACKAGE			
	PINS	PIN POSITION	MATERIAL	CODE
PCA82C250	8	DIL8	plastic	SOT97AF8
PCA82C250T	8	SO8	plastic	SOT96AG4

CAN controller interface

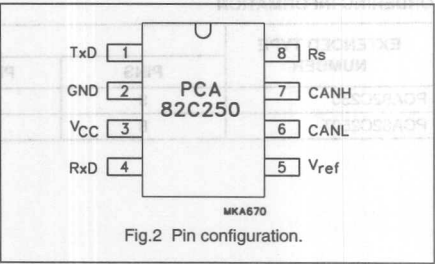
PCA82C250

BLOCK DIAGRAM



PINNING

SYMBOL	PIN	DESCRIPTION
TxD	1	transmit data input
GND	2	ground
V <sub>CC</sub>	3	supply voltage
RxD	4	receive data output
V <sub>ref</sub>	5	reference voltage output
CANL	6	LOW level CAN voltage input/output
CANH	7	HIGH level CAN voltage input/output
Rs	8	slope resistor input





CAN controller interface

PCA82C250

FUNCTIONAL DESCRIPTION

The PCA82C250 is the interface between the CAN protocol controller and the physical bus. It is primarily intended for high-speed applications (up to 1 Mbaud) in cars. The device provides differential transmit capability to the bus and differential receive capability to the CAN controller. It is fully compatible with the ISO/DIS 11898 standard.

A current limiting circuit protects the transmitter output stage against short-circuit to positive and negative battery voltage. Although the power dissipation is increased during this fault condition, this feature will prevent destruction of the transmitter output stage.

If the junction temperature exceeds a value of approximately 160 °C, the limiting current of both transmitter outputs is decreased. Because the transmitter is responsible for the major part of the power dissipation, this will result in a reduced power dissipation and hence a lower chip temperature. All other parts of the IC will remain in operation. The thermal protection is particularly needed when a bus line is short-circuited.

The CANH and CANL lines are also protected against electrical transients which may occur in an automotive environment.

Pin 8 (Rs) allows three different modes of operation to be selected: high-speed, slope control or standby.

For high-speed operation, the transmitter output transistors are simply switched on and off as fast as possible. In this mode, no measures are taken to limit the rise and fall slope. Use of a shielded cable is recommended to avoid RFI problems. The high-speed mode is selected by connecting pin 8 to ground.

For lower speeds or shorter bus length, an unshielded twisted pair or a parallel pair of wires can be used for the bus. To reduce RFI, the rise and fall slope should be limited. The rise and fall slope can be programmed with a resistor connected from pin 8 to ground. The slope is proportional to the current output at pin 8.

If a HIGH level is applied, the circuit enters a low current standby mode. In this mode, the transmitter is switched off and the receiver is switched to a low current. If dominant bits are detected (differential bus voltage > 0.9 V), RxD will be switched to a LOW level. The microcontroller should react to this condition by switching the transceiver back to normal operation (via pin 8). Because the receiver is slow in standby mode, the first message will be lost.

Table 1 Truth table of CAN transceiver.

SUPPLY	TxD	CANH	CANL	BUS STATE	RxD
4.5 to 5.5 V	0	HIGH	LOW	dominant	0
4.5 to 5.5 V	1 (or floating)	floating	floating	recessive	1
< 2 V (not powered)	X	floating	floating	recessive	X
2 V < V <sub>CC</sub> < 4.5 V	> 0.75V <sub>CC</sub>	floating	floating	recessive	X
2 V < V <sub>CC</sub> < 4.5 V	X	floating if V <sub>Rs</sub> > 0.75V <sub>CC</sub>	floating if V <sub>Rs</sub> > 0.75V <sub>CC</sub>	recessive	X

Table 2 Rs (pin 8) summary.

CONDITION FORCED AT Rs	MODE	RESULTING VOLTAGE OR CURRENT AT Rs
V <sub>Rs</sub> > 0.75V <sub>CC</sub>	standby	I <sub>Rs</sub> < 110 µA
10 µA < -I <sub>Rs</sub> < 200 µA	slope control	0.4V <sub>CC</sub> < V <sub>Rs</sub> < 0.6V <sub>CC</sub>
V <sub>Rs</sub> < 0.3V <sub>CC</sub>	high-speed	-I <sub>Rs</sub> < 500 µA

CAN controller interface

PCA82C250

LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134). All voltages are referenced to pin 2; positive input current.

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V <sub>CC</sub>	supply voltage		-0.3	+9.0	V
V <sub>I</sub>	DC voltage at pins 1, 4, 5 and 8		-0.3	V <sub>CC</sub> + 0.3	V
V <sub>6,7</sub>	DC voltage at pins 6 and 7	0 V < V <sub>CC</sub> < 5.5 V; no time limit	-8.0	+18.0	V
V <sub>trt</sub>	transient voltage at pins 6 and 7	see Fig.8	-150	+100	V
T <sub>stg</sub>	storage temperature		-55	+150	°C
T <sub>amb</sub>	operating ambient temperature		-40	+125	°C
T <sub>vj</sub>	virtual junction temperature	note 1	-40	+150	°C

Note

1. In accordance with IEC 747-1:  
An alternative definition of virtual junction temperature T<sub>vj</sub> is:  
 $T_{vj} = T_{amb} + P_d \times R_{th\ vj-amb}$   
where R<sub>th vj-amb</sub> is a fixed value to be used for the calculation of T<sub>vj</sub>.  
The rating for T<sub>vj</sub> limits the allowable combinations of power dissipation (P<sub>d</sub>) and ambient temperature (T<sub>amb</sub>).

HANDLING

Classification A: human body model: C = 100 pF; R = 1500 Ω; V = ±2000 V.  
Classification B: machine model: C = 200 pF; R = 0 Ω; V = ±200 V.

THERMAL RESISTANCE

SYMBOL	PARAMETER	THERMAL RESISTANCE
R <sub>th ja</sub>	from junction to ambient in free air	
	PCA82C250	100 K/W
	PCA82C250T	160 K/W

MODE	CONDITION FORCED AT R <sub>s</sub>	RESULTING VOLTAGE OR CURRENT AT R <sub>s</sub>
standby	V <sub>CE</sub> > 0.75V <sub>CC</sub>	I <sub>AS</sub> > 17 mA
slope control	0.75V <sub>CC</sub> < V <sub>CE</sub> < 0.9V <sub>CC</sub>	
high speed	V <sub>CE</sub> > 0.9V <sub>CC</sub>	I <sub>AS</sub> > 200 mA

## CAN controller interface

## PCA82C250

## CHARACTERISTICS

$V_{CC} = 4.5$  to  $5.5$  V;  $T_{amb} = -40$  to  $+125$  °C;  $R_L = 60$   $\Omega$ ;  $I_B = > -10$   $\mu$ A; unless otherwise specified. All voltages referenced to ground (pin 2); positive input current; all parameters are guaranteed over the ambient temperature range by design, but only 100% tested at  $+25$  °C.

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
<b>Supply</b>						
$I_3$	supply current	dominant; $V_1 = 1$ V	—	—	70	mA
		recessive; $V_1 = 4$ V; $R_8 = 47$ k $\Omega$	—	—	14	mA
		recessive; $V_1 = 4$ V; $V_8 = 1$ V	—	—	18	mA
		standby; $T_{amb} < 90$ °C; note 1	—	100	170	$\mu$ A
<b>DC bus transmitter</b>						
$V_{IH}$	HIGH level input voltage	output recessive	$0.7V_{CC}$	—	$V_{CC} + 0.3$	V
$V_{IL}$	LOW level input voltage	output dominant	-0.3	—	$0.3V_{CC}$	V
$I_{IH}$	HIGH level input current	$V_1 = 4$ V	-200	—	+30	$\mu$ A
$I_{IL}$	LOW level input current	$V_1 = 1$ V	100	—	600	$\mu$ A
$V_{6,7}$	recessive bus voltage	$V_1 = 4$ V; no load	2.0	—	3.0	V
$I_{LO}$	off-state output leakage current	$-2$ V $< (V_6, V_7) < 7$ V	-2	—	+1	mA
		$-2$ V $< (V_6, V_7) < 7$ V	-5	—	+12	mA
$V_7$	CANH output voltage	$V_1 = 1$ V	2.75	—	4.5	V
$V_6$	CANL output voltage	$V_1 = 1$ V	0.5	—	2.25	V
$\Delta V_{6,7}$	difference between output voltage at pins 6 and 7	$V_1 = 1$ V	1.5	—	3.0	V
		$V_1 = 1$ V; $R_L = 45$ $\Omega$ ; $V_{CC} = 4.9$ V	1.5	—	—	V
		$V_1 = 4$ V; no load	-500	—	+50	mA
$I_{sc7}$	short-circuit CANH current	$V_7 = -5$ V; $V_{CC} \leq 5$ V	—	—	105	mA
		$V_7 = -5$ V; $V_{CC} = 5.5$ V	—	—	120	mA
$I_{sc6}$	short-circuit CANL current	$V_6 = +18$ V	—	—	160	mA
<b>DC bus receiver: <math>V_1 = 4</math> V; pins 6 and 7 externally driven; <math>-2</math> V <math>&lt; (V_6, V_7) &lt; 7</math> V</b>						
$V_{diff(r)}$	differential input voltage (recessive)		-1.0	—	0.5	V
		$-7$ V $< (V_6, V_7) < 12$ V; not standby mode	-1.0	—	0.4	V
$V_{diff(d)}$	differential input voltage (dominant)		0.9	—	5.0	V
		$-7$ V $< (V_6, V_7) < 12$ V; not standby mode	1.0	—	5.0	V
$V_{diff(hys)}$	differential input hysteresis	see Fig.5	—	150	—	mV
$V_{OH}$	HIGH level output voltage (pin 4)	$I_4 = -100$ $\mu$ A	$0.8V_{CC}$	—	$V_{CC}$	V
$V_{OL}$	LOW level output voltage (pin 4)	$I_4 = 1$ mA	0	—	$0.2V_{CC}$	V
		$I_4 = 10$ mA	0	—	1.5	V

## CAN controller interface

## PCA82C250

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
$R_i$	CANH, CANL input resistance		5	—	25	k $\Omega$
$R_{diff}$	differential input resistance		20	—	100	k $\Omega$
$C_i$	CANH, CANL input capacitance		—	—	20	pF
$C_{diff}$	differential input capacitance		—	—	10	pF
<b>Reference output</b>						
$V_{ref}$	reference output voltage	$V_8 = 1\text{ V};$ $-50\text{ }\mu\text{A} < I_5 < 50\text{ }\mu\text{A}$	$0.45V_{CC}$	—	$0.55V_{CC}$	V
		$V_8 = 4\text{ V};$ $-50\text{ }\mu\text{A} < I_5 < 50\text{ }\mu\text{A}$	$0.4V_{CC}$	—	$0.6V_{CC}$	V
<b>Timing (see Figs 3, 4, 6 and 7)</b>						
$t_{bit}$	minimum bit time	$V_8 = 1\text{ V}$	—	—	1	$\mu\text{s}$
$t_{onTxD}$	delay TxD to bus active	$V_8 = 1\text{ V}$	—	—	50	ns
$t_{offTxD}$	delay TxD to bus inactive	$V_8 = 1\text{ V}$	—	40	80	ns
$t_{onRxD}$	delay TxD to receiver active	$V_8 = 1\text{ V}$	—	55	120	ns
$t_{offRxD}$	delay TxD to receiver inactive	$V_8 = 1\text{ V}; V_{CC} < 5.1\text{ V};$ $T_{amb} = < +85\text{ }^\circ\text{C}$	—	82	150	ns
		$V_8 = 1\text{ V}; V_{CC} < 5.1\text{ V};$ $T_{amb} = < +125\text{ }^\circ\text{C}$	—	82	170	ns
		$V_8 = 1\text{ V}; V_{CC} < 5.5\text{ V};$ $T_{amb} = < +85\text{ }^\circ\text{C}$	—	90	170	ns
		$V_8 = 1\text{ V}; V_{CC} < 5.5\text{ V};$ $T_{amb} = < +125\text{ }^\circ\text{C}$	—	90	190	ns
$t_{onRxD}$	delay TxD to receiver active	$R_8 = 47\text{ k}\Omega$	—	390	520	ns
		$R_8 = 24\text{ k}\Omega$	—	260	320	ns
$t_{offRxD}$	delay TxD to receiver inactive	$R_8 = 47\text{ k}\Omega$	—	260	450	ns
		$R_8 = 24\text{ k}\Omega$	—	210	320	ns
ISRI	differential output voltage slew rate	$R_8 = 47\text{ k}\Omega$	—	14	—	V/ $\mu\text{s}$
$t_{WAKE}$	wake up time from standby (via pin 8)		—	—	20	$\mu\text{s}$
$t_{dRxD}$	bus dominant to RxD LOW	$V_8 = 4\text{ V};$ standby mode	—	—	3	$\mu\text{s}$
<b>Standby/slope control (pin 8)</b>						
$V_8$	input voltage for high-speed		—	—	$0.3V_{CC}$	V
$I_8$	input current for high-speed	$V_8 = 1\text{ V}$	—	—	-500	$\mu\text{A}$

CAN controller interface

PCA82C250

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
$V_{\text{stb}}$	input voltage for standby mode		$0.75V_{\text{CC}}$	—	—	V
$I_{\text{slope}}$	slope control mode current		−10	—	−200	$\mu\text{A}$
$V_{\text{slope}}$	slope control mode voltage		$0.4V_{\text{CC}}$	—	$0.6V_{\text{CC}}$	V

Note

1.  $I_1 = I_4 = I_5 = 0 \text{ mA}$ ;  $0 < V_7$ ;  $V_6 < V_{\text{CC}}$ ;  $V_8 = V_{\text{CC}}$ .

QUALITY SPECIFICATION

Quality in accordance with SNW - FQ 611 - part-E.

TEST AND APPLICATION INFORMATION

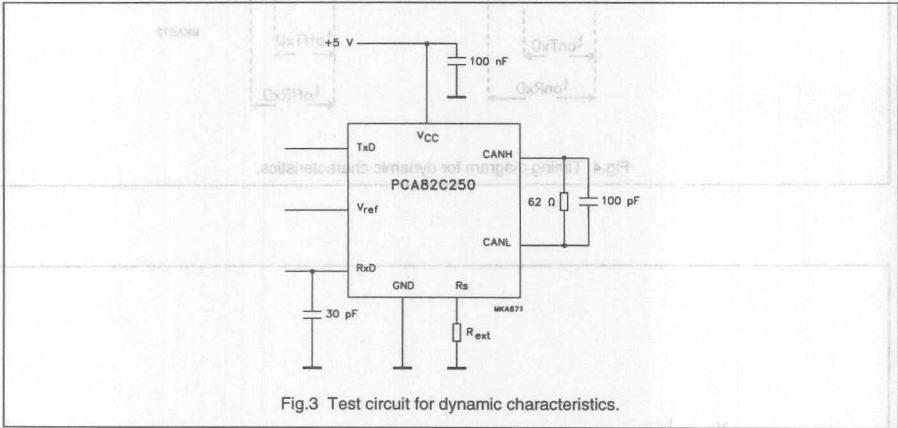
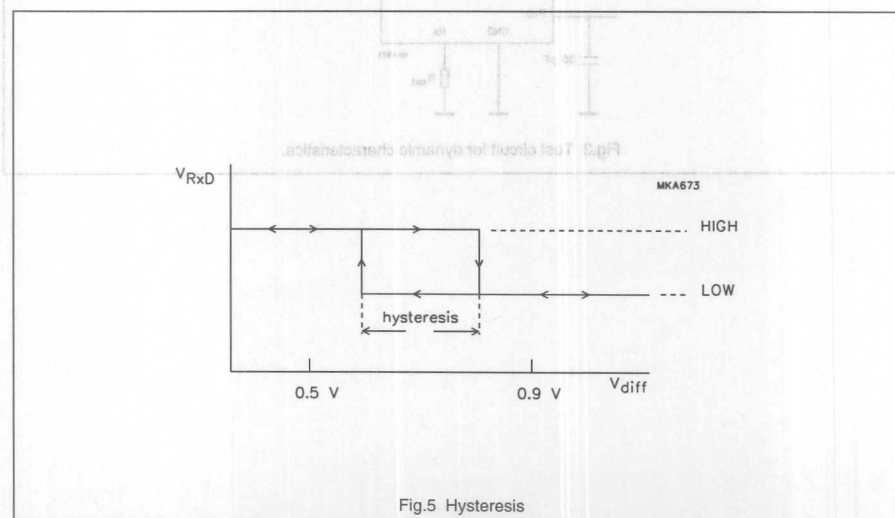
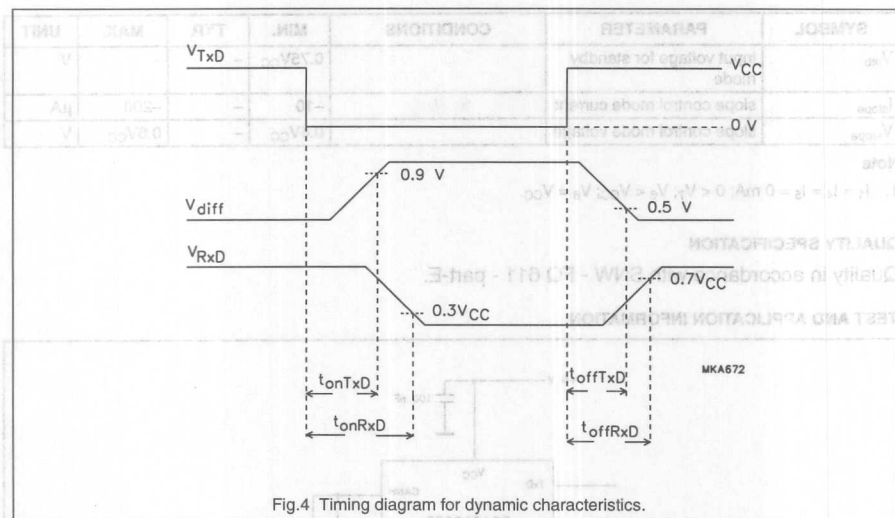


Fig.3 Test circuit for dynamic characteristics.





CAN controller interface

PCA82C250

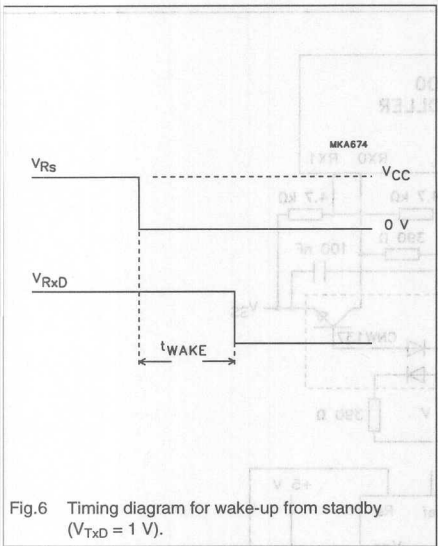


Fig.6 Timing diagram for wake-up from standby ( $V_{TxD} = 1\text{ V}$ ).

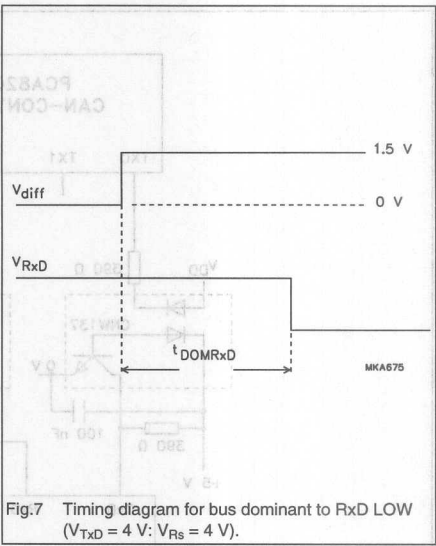


Fig.7 Timing diagram for bus dominant to RxD LOW ( $V_{TxD} = 4\text{ V}$ ;  $V_{Rs} = 4\text{ V}$ ).

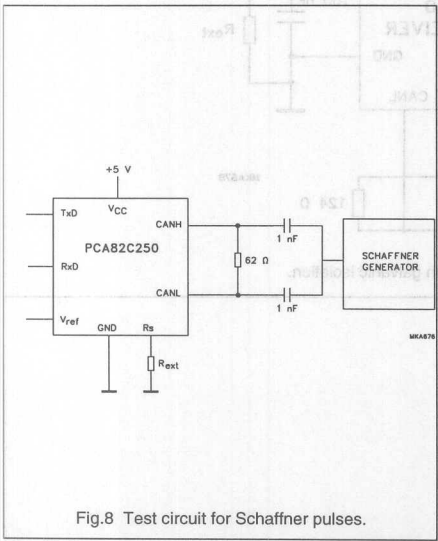


Fig.8 Test circuit for Schaffner pulses.

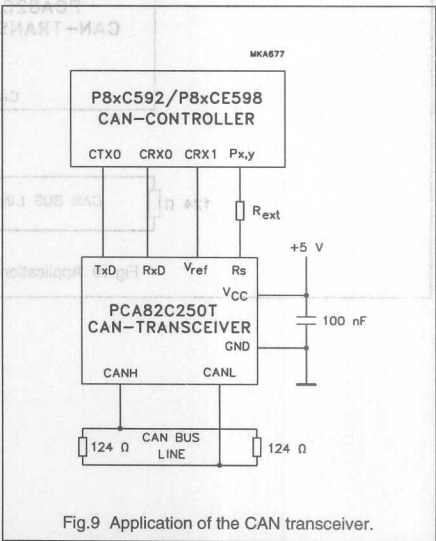


Fig.9 Application of the CAN transceiver.

## CAN controller interface

## PCA82C250

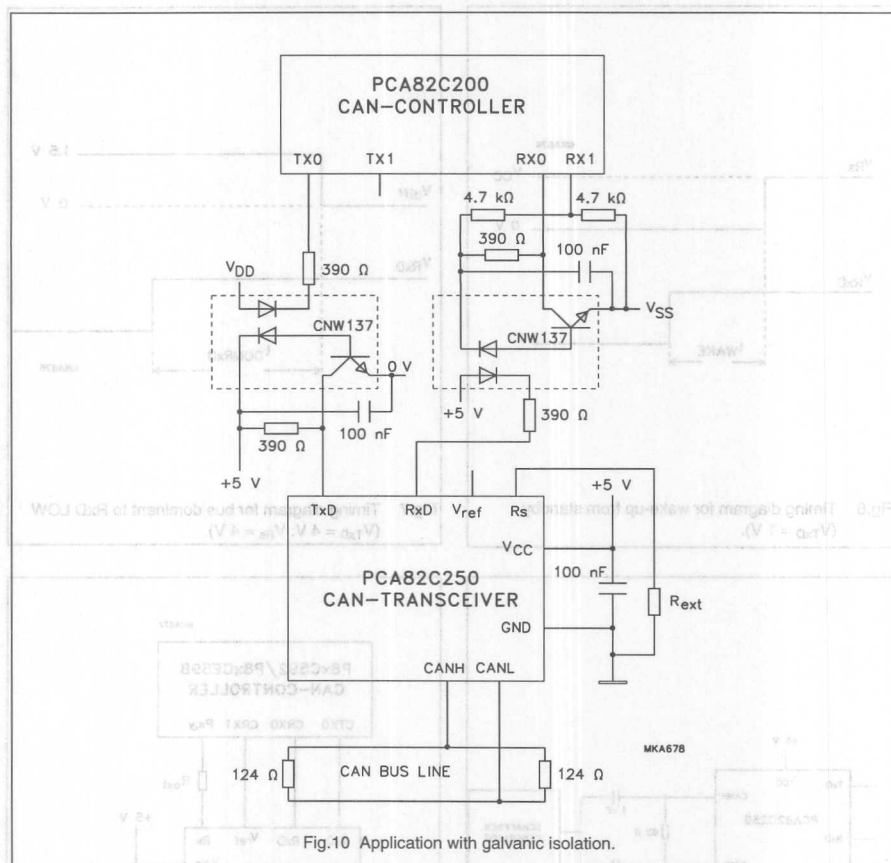
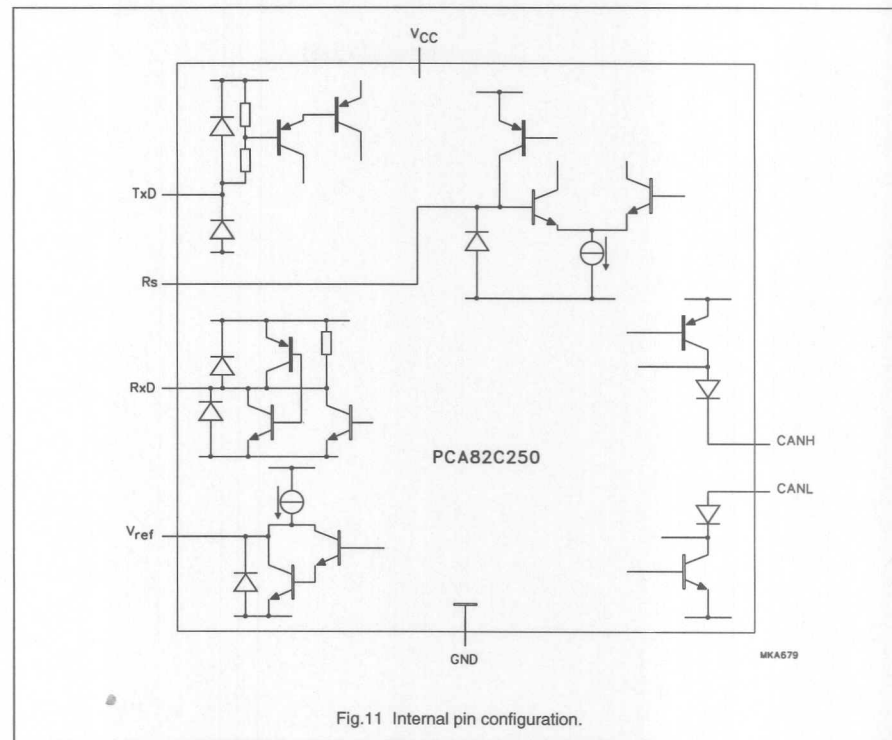


Fig.10 Application with galvanic isolation.

## CAN controller interface

## PCA82C250







# Section 7

## Other Microcontrollers

### 80C51-Based 8-Bit Microcontrollers

CONTENTS

8031AH/8051AH	Single-chip 8-bit microcontroller .....	1257
8032AH/8052AH	Single-chip 8-bit microcontroller .....	1266



## Single-chip 8-bit microcontroller

8031AH/8051AH

## DESCRIPTION

The Philips 8031AH/8051AH is a high-performance microcontroller fabricated with Philips high-density highly reliable +5V, depletion-load, N-channel, silicon-gate, N500 MOS process technology. It provides the hardware features, architectural enhancements and instructions that are necessary to make it a powerful and cost-effective controller for applications requiring up to 64k bytes of program memory and/or up to 64k bytes of data storage.

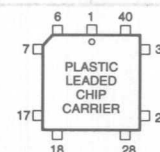
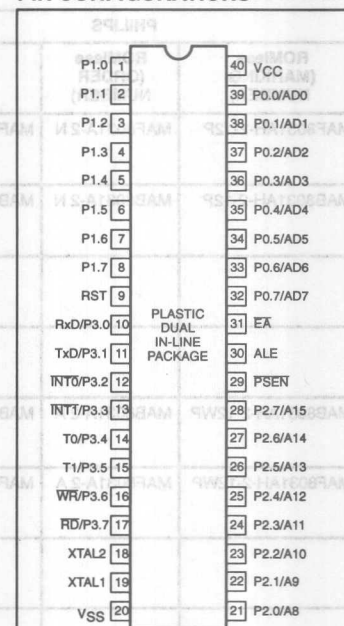
The 8051AH contains a 4k × 8 read-only program memory, a 128 × 8 read-only data memory, 32 I/O lines, two 16-bit counter/timers, a five-source, two-priority level nested interrupt structure, a serial I/O port for either multi-processor communications, I/O expansion or full duplex UART, and on-chip oscillator and clock circuits. The 8031AH is identical, except that it lacks the program memory. For systems that require extra capability, the 8051AH can be expanded using standard TTL compatible memories and byte oriented peripheral controllers.

The 8051AH microcontroller, like its 8048 predecessor, is efficient both as a controller and as an arithmetic processor. It has extensive facilities for binary and BCD arithmetic and excels in bit-handling capabilities. Efficient use of program memory results from an instruction set consisting of 44% one-byte, 41% two-byte, and 15% three-byte instructions. With a 12MHz crystal, 58% of the instructions execute in 1μs, 40% in 2μs and multiply and divide require only 4μs.

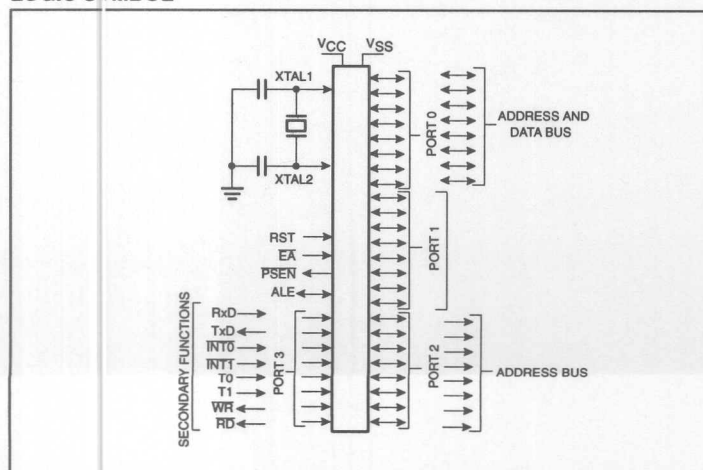
## FEATURES

- Reduced supply current
- 4k × 8 ROM (8051AH)
- 128 × 8 RAM
- Four 8-bit ports, 32 I/O lines
- Two 16-bit timer/event counters
- High-performance full-duplex serial channel
- External memory expandable to 128k
- Boolean processor
- Industry standard 8051 architecture:
  - Non-paged jumps
  - Direct addressing
  - Four 8-register banks
  - Stack depth up to 128-bytes
  - Multiply, divide, subtract, compare
- Most instructions execute in 1μs
- 4μs multiply and divide

## PIN CONFIGURATIONS



## LOGIC SYMBOL



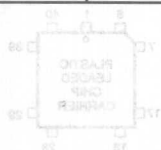
Pin	Function	Pin	Function	Pin	Function
1	NC	15	INT1/P3.3	30	P2.6/A14
2	P1.0	16	T0/P3.4	31	P2.7/A15
3	P1.1	17	T1/P3.5	32	PSEN
4	P1.2	18	WR/P3.6	33	ALE
5	P1.3	19	RD/P3.7	34	NC
6	P1.4	20	XTAL2	35	EA
7	P1.5	21	XTAL1	36	P0.7/AD7
8	P1.6	22	VSS	37	P0.6/AD6
9	P1.7	23	NC	38	P0.5/AD5
10	RST	24	P2.0/A8	39	P0.4/AD4
11	RxD/P3.0	25	P2.1/A9	40	P0.3/AD3
12	NC	26	P2.2/A10	41	P0.2/AD2
13	TxD/P3.1	27	P2.3/A11	42	P0.1/AD1
14	INT0/P3.2	28	P2.4/A12	43	P0.0/AD0
		29	P2.5/A13	44	VCC

## Single-chip 8-bit microcontroller

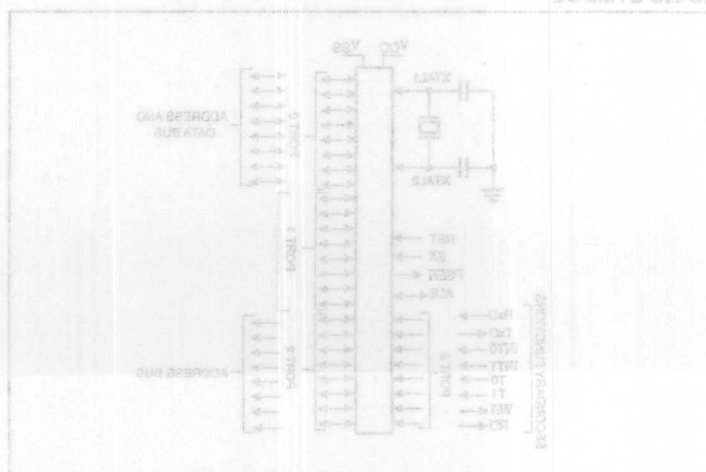
## 8031AH/8051AH

## PART NUMBER SELECTION

PHILIPS			PHILIPS NORTH AMERICA		TEMPERATURE °C AND PACKAGE	FREQ. MHz	DRAWING NUMBER
ROMless (MARKING NUMBER)	ROMless (ORDER NUMBER)	ROM	ROMless	ROM			
MAF8031AH-2-12P	MAF8031A-2 N	MAF8051AH-2P	SCN8031HACN40	SCN8051HACN40	-40 to +85, Plastic Dual In-Line Package	12	0415C
MAB8031AH-2-12P	MAB8031A-2 N	MAB8051AH-2P	SCN8031HCCN40	SCN8051HCCN40	0 to +70, Plastic Dual In-Line Package	12	0415C
			SCN8031HCFN40	SCN8051HCFN40	0 to +70, Plastic Dual In-Line Package	15	0415C
			SCN8031HAFN40	SCN8051HAFN40	-40 to +85, Plastic Dual In-Line Package	15	0415C
MAB8031AH-2-12WP	MAB8031A-2 A	MAB8051AH-2WP	SCN8031HCCA44	SCN8051HCCA44	0 to +70, Plastic Leaded Chip Carrier	12	0403G
MAF8031AH-2-12WP	MAF8031A-2 A	MAF8051AH-2WP	SCN8031HACA44	SCN8051HACA44	-40 to +85, Plastic Leaded Chip Carrier	12	0403G
			SCN8031HCFA44	SCN8051HCFA44	0 to +70, Plastic Leaded Chip Carrier	15	0403G
			SCN8031HAFA44	SCN8051HAFA44	-40 to +85, Plastic Leaded Chip Carrier	15	0403G



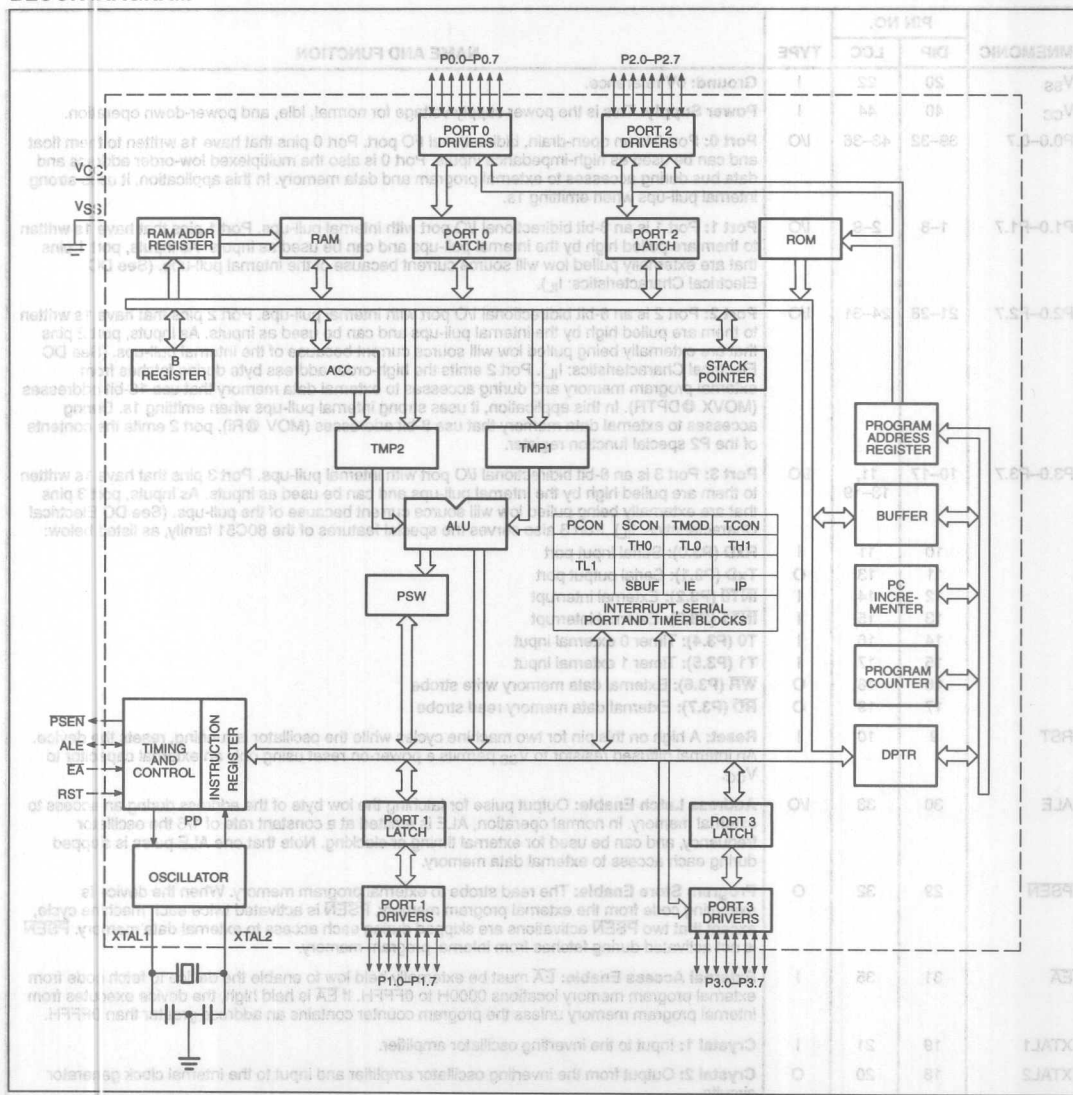
Pin	Function	Pin	Function	Pin	Function	Pin	Function
1	P0	17	P16	33	P32	49	P40
2	P0	18	P17	34	P33	50	P41
3	P0	19	P18	35	P34	51	P42
4	P0	20	P19	36	P35	52	P43
5	P0	21	P20	37	P36	53	P44
6	P0	22	P21	38	P37	54	P45
7	P0	23	P22	39	P38	55	P46
8	P0	24	P23	40	P39	56	P47
9	P0	25	P24			57	P48
10	P0	26	P25			58	P49
11	P0	27	P26			59	P50
12	P0	28	P27			60	P51
13	P0	29	P28			61	P52
14	P0	30	P29			62	P53
15	P0	31	P30			63	P54
16	P0	32	P31			64	P55



## Single-chip 8-bit microcontroller

8031AH/8051AH

## BLOCK DIAGRAM



high and low times specified in the data sheet must be observed.

**DESIGN CONSIDERATIONS**  
 At power-on, the voltage on VCC and RST should come up at the same time for a proper start-up.

To drive the device from an external clock source, XTAL2 should be driven with XTAL1. XTAL1 is connected to ground. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum

**OSCILLATOR CHARACTERISTICS**  
 XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator, as shown in the logic symbol.



## Single-chip 8-bit microcontroller

8031AH/8051AH

## PIN DESCRIPTIONS

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	LCC		
V <sub>SS</sub>	20	22	I	<b>Ground:</b> 0V reference.
V <sub>CC</sub>	40	44	I	<b>Power Supply:</b> This is the power supply voltage for normal, idle, and power-down operation.
P0.0–P0.7	39–32	43–36	I/O	<b>Port 0:</b> Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s.
P1.0–P1.7	1–8	2–9	I/O	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ).
P2.0–P2.7	21–28	24–31	I/O	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register.
P3.0–P3.7	10–17	11, 13–19	I/O	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: I <sub>IL</sub> ). Port 3 also serves the special features of the 80C51 family, as listed below:
	10	11	I	<b>RxD (P3.0):</b> Serial input port
	11	13	O	<b>TxD (P3.1):</b> Serial output port
	12	14	I	<b>INT0 (P3.2):</b> External interrupt
	13	15	I	<b>INT1 (P3.3):</b> External interrupt
	14	16	I	<b>T0 (P3.4):</b> Timer 0 external input
	15	17	I	<b>T1 (P3.5):</b> Timer 1 external input
	16	18	O	<b>WR (P3.6):</b> External data memory write strobe
	17	19	O	<b>RD (P3.7):</b> External data memory read strobe
RST	9	10	I	<b>Reset:</b> A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to V <sub>SS</sub> permits a power-on reset using only an external capacitor to V <sub>CC</sub> .
ALE	30	33	I/O	<b>Address Latch Enable:</b> Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory.
PSEN	29	32	O	<b>Program Store Enable:</b> The read strobe to external program memory. When the device is executing code from the external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory. PSEN is not activated during fetches from internal program memory.
EA	31	35	I	<b>External Access Enable:</b> EA must be externally held low to enable the device to fetch code from external program memory locations 0000H to 0FFFH. If EA is held high, the device executes from internal program memory unless the program counter contains an address greater than 0FFFH.
XTAL1	19	21	I	<b>Crystal 1:</b> Input to the inverting oscillator amplifier.
XTAL2	18	20	O	<b>Crystal 2:</b> Output from the inverting oscillator amplifier and input to the internal clock generator circuits.

## OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator, as shown in the logic symbol.

To drive the device from an external clock source, XTAL2 should be driven while XTAL1 is connected to ground. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum

high and low times specified in the data sheet must be observed.

## DESIGN CONSIDERATIONS

At power-on, the voltage on V<sub>CC</sub> and RST should come up at the same time for a proper start-up.

## Single-chip 8-bit microcontroller

8031AH/8051AH

ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>

PARAMETER	RATING	UNIT
Storage temperature range	-65 to +150	°C
Voltage on any other pin to $V_{SS}$	-0.5 to +7.0	V
Input, output current on any single pin	10	mA
Power dissipation	1.0	W

## DC ELECTRICAL CHARACTERISTICS

 $T_{amb} = 0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}^{4, 5}$ 

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			MIN	MAX	
$V_{IL}$	Input low voltage		-0.5	0.8	V
$V_{IH}$	Input high voltage; except XTAL2, RST		2.0	$V_{CC}+0.5$	V
$V_{IH1}$	Input high voltage to RST for reset, XTAL2	XTAL1 to $V_{SS}$	2.5	$V_{CC}+0.5$	V
$V_{OL}$	Output low voltage; ports 1, 2, 3 <sup>6</sup>	$I_{OL} = 1.6\text{mA}$		0.45	V
$V_{OL1}$	Output low voltage; port 0, ALE, $\overline{\text{PSEN}}^6$	$I_{OL} = 3.2\text{mA}$		0.45	V
$V_{OH}$	Output high voltage; ports 1, 2, 3	$I_{OH} = -80\mu\text{A}$	2.4		V
$V_{OH1}$	Output high voltage; port 0, ALE, $\overline{\text{PSEN}}^3$	$I_{OH} = -400\mu\text{A}$	2.4		V
$I_{IL}$	Logical 0 input current; ports 1, 2, 3	$V_{IN} = 0.45\text{V}$		-500	$\mu\text{A}$
$I_{IH1}$	Input high current to RST for reset	$V_{IN} < V_{CC} - 1.5\text{V}$		500	$\mu\text{A}$
$I_{LI}$	Input leakage current; port 0, $\overline{\text{EA}}$	$0.45 < V_{IN} < V_{CC}$		$\pm 10$	$\mu\text{A}$
$I_{IL2}$	Logical 0 input current for XTAL2	XTAL1 = $V_{SS}$ , $V_{IN} = 0.45\text{V}$		-3.2	mA
$I_{CC}$	Power supply current	All outputs disconnected and $\overline{\text{EA}} = V_{CC}$		125	mA
$C_{IO}$	Pin capacitance			10	pF

 $T_{amb} = -40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$ 

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			MIN	MAX	
$V_{IH}$	Input high voltage; except XTAL2, RST		2.1	$V_{CC}+0.5$	V
$V_{IH1}$	Input high voltage to RST and XTAL2	XTAL1 = $V_{SS}$	2.6	$V_{CC}+0.5$	V
$I_{IL2}$	Logical 0 input current for XTAL2	XTAL1 = $V_{SS}$ , $V_{IN} = 0.45\text{V}$		-4.0	mA
$I_{CC}$	Power supply current	All outputs disconnected and $\overline{\text{EA}} = V_{CC}$		135	mA

## NOTES:

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
- For operating at elevated temperatures, the device must be derated based on  $+150^{\circ}\text{C}$  maximum junction temperature.
- This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
- Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to  $V_{SS}$  unless otherwise noted.
- All voltage measurements are referenced to ground. For testing, all input signals swing between 0.45V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and at output voltages of 0.8V and 2.0V as appropriate.
- $V_{OL}$  is derated when the device rapidly discharges external capacitance. This AC noise is most pronounced during emission of address data. When using external memory, locate the latch or buffer as close as possible to the device.

	Emitted Ports	Degraded I/O Lines	$V_{OL}$ (Peak Max)
Data In	P2, P0	P1, P3	0.8V
Address	P0	P1, p3, ALE	0.8V
Write Data	P0	P1, p3, ALE	0.8V

- $C_L = 10\text{pF}$  for port 0, ALE and  $\overline{\text{PSEN}}$  outputs;  $C_L = 80\text{pF}$  for all other ports.

## Single-chip 8-bit microcontroller

8031AH/8051AH

## AC ELECTRICAL CHARACTERISTICS

 $T_{amb} = 0^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$  or  $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ ,  $V_{CC} = 5\text{V} \pm 20\%$ ,  $V_{SS} = 0\text{V}^{1,2}$ 

SYMBOL	FIGURE	PARAMETER	12MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
$1/t_{CLCL}$		Oscillator frequency: Speed Versions SCN8051/31 C MAB8051/31 -2 MAF8051/31 -2 SCN8051/31 F			3.5 3.5 3.5 3.5	12 12 12 15	MHz
$t_{LHLL}$	1	ALE pulse width	127		$2t_{CLCL}-40$		ns
$t_{AVLL}$	1	Address valid to ALE low	43		$t_{CLCL}-40$		ns
$t_{LLAX}$	1	Address hold after ALE low	48		$t_{CLCL}-35$		ns
$t_{LLIV}$	1	ALE low to valid instruction in		233		$4t_{CLCL}-100$	ns
$t_{LLPL}$	1	ALE low to PSEN low	58		$t_{CLCL}-25$		ns
$t_{PLPH}$	1	PSEN pulse width	215		$3t_{CLCL}-35$		ns
$t_{PLIV}$	1	PSEN low to valid instruction in		125		$3t_{CLCL}-125$	ns
$t_{PXIX}$	1	Input instruction hold after PSEN	0		0		ns
$t_{PXIZ}$	1	Input instruction float after PSEN		63		$t_{CLCL}-20$	ns
$t_{AVIV}$	1	Address to valid instruction in		302		$5t_{CLCL}-115$	ns
$t_{PLAZ}$	1	PSEN low to address float		20		20	ns
$t_{PXAV}$	1	PSEN to address valid	75		$t_{CLCL}-8$		ns
<b>Data Memory</b>							
$t_{RLRH}$	2, 3	RD pulse width	400		$6t_{CLCL}-100$		ns
$t_{WLWH}$	2, 3	WR pulse width	400		$6t_{CLCL}-100$		ns
$t_{RLDV}$	2, 3	RD low to valid data in		252		$5t_{CLCL}-165$	ns
$t_{RHDX}$	2, 3	Data hold after RD	0		0		ns
$t_{RHDZ}$	2, 3	Data float after RD		97		$2t_{CLCL}-70$	ns
$t_{LLDV}$	2, 3	ALE low to valid data in		517		$8t_{CLCL}-150$	ns
$t_{AVDV}$	2, 3	Address to valid data in		585		$9t_{CLCL}-165$	ns
$t_{LLWL}$	2, 3	ALE low to RD or WR low	200	300	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
$t_{AVWL}$	2, 3	Address valid to WR low or RD low	203		$4t_{CLCL}-130$		ns
$t_{QVWX}$	2, 3	Data valid to WR transition	23		$t_{CLCL}-60$		ns
$t_{QVWH}$	2, 3	Data valid to WR high	433		$7t_{CLCL}-150$		ns
$t_{WHQX}$	2, 3	Data hold after WR	33		$t_{CLCL}-50$		ns
$t_{RLAZ}$	2, 3	RD low to address float		20		20	ns
$t_{WHLH}$	2, 3	RD or WR high to ALE high	43	123	$t_{CLCL}-40$	$t_{CLCL}+40$	ns
<b>External Clock</b>							
$t_{CHCX}$	5	High time	20		20		ns
$t_{CLCX}$	5	Low time	20		20		ns
$t_{CLCH}$	5	Rise time		20		20	ns
$t_{CHCL}$	5	Fall time		20		20	ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.

## Single-chip 8-bit microcontroller

8031AH/8051AH

### EXPLANATION OF THE AC SYMBOLS

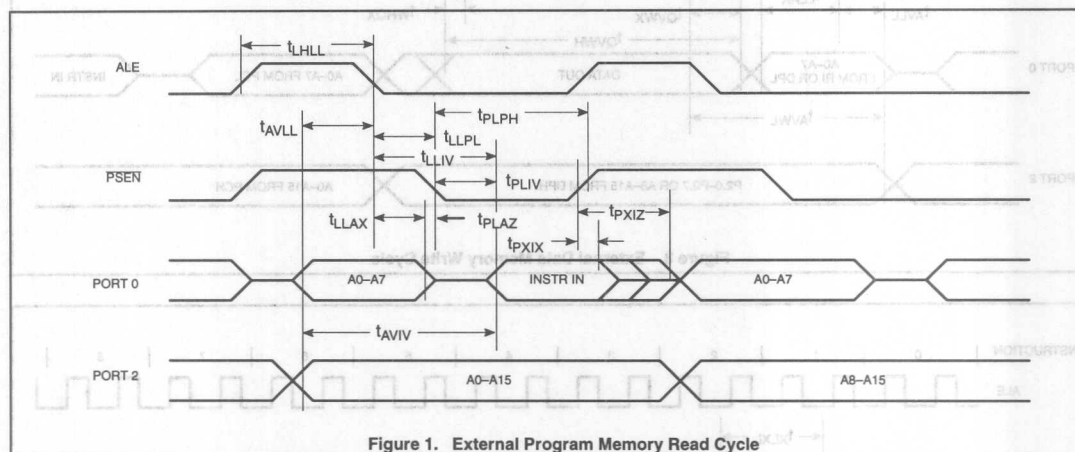
Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

- A – Address
- C – Clock
- D – Input data
- H – Logic level high
- I – Instruction (program memory contents)
- L – Logic level low, or ALE
- P – PSEN

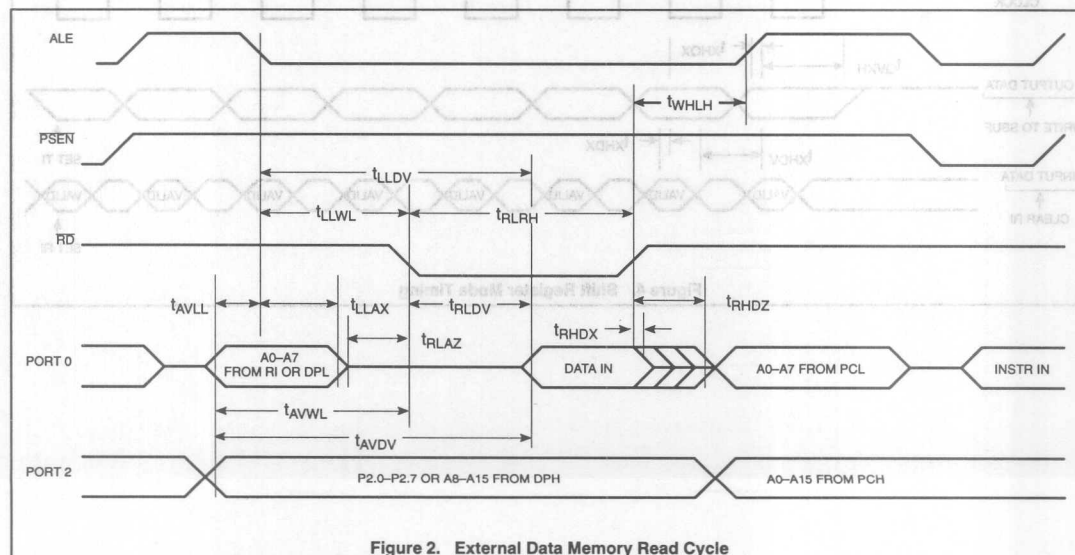
Q – Output data  
R –  $\overline{RD}$  signal  
t – Time  
V – Valid  
W –  $\overline{WR}$  signal  
X – No longer a valid logic level  
Z – Float

**Examples:**  $t_{AVLL}$  = Time for address valid to ALE low.

$t_{LLPL}$  = Time for ALE low to  $\overline{PSEN}$  low.



**Figure 1. External Program Memory Read Cycle**



**Figure 2. External Data Memory Read Cycle**

## Single-chip 8-bit microcontroller

8031AH/8051AH

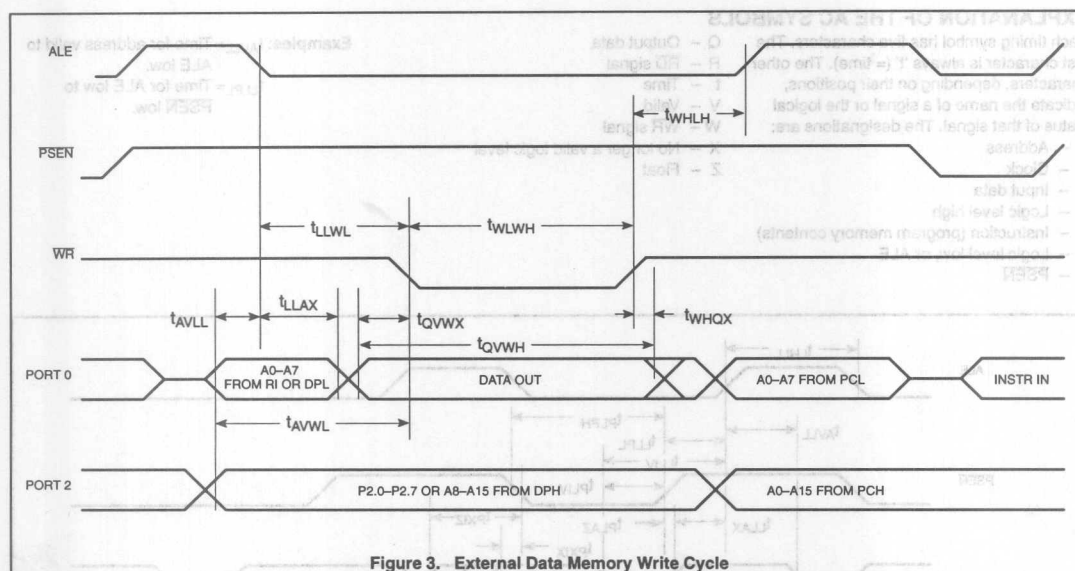


Figure 3. External Data Memory Write Cycle

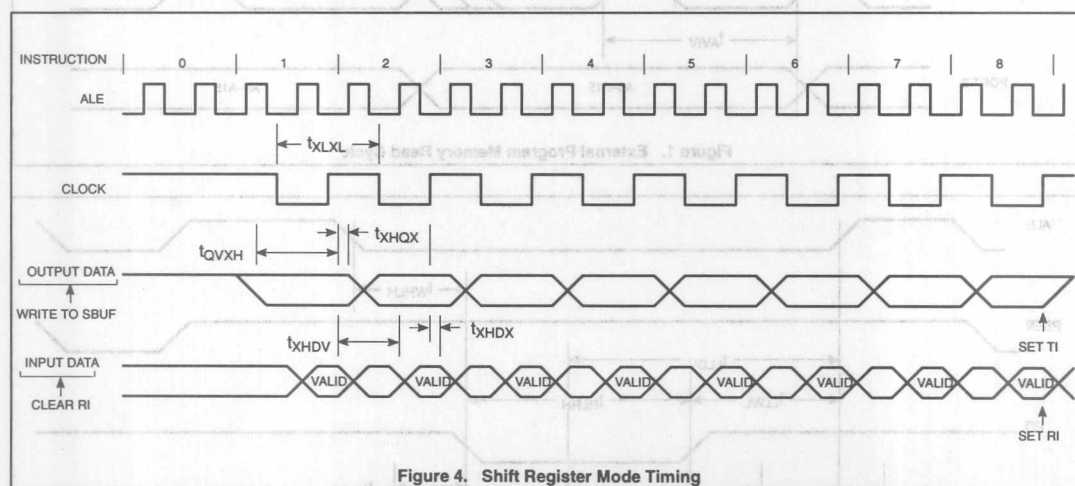
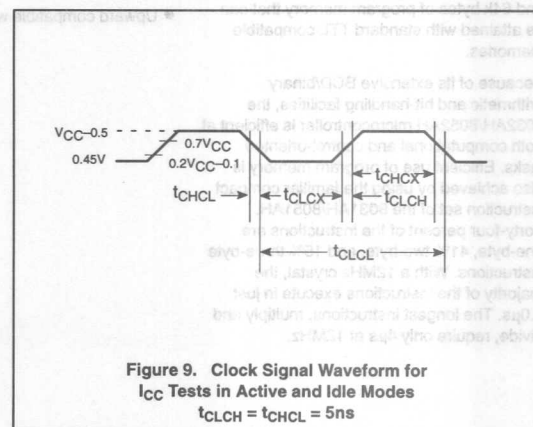
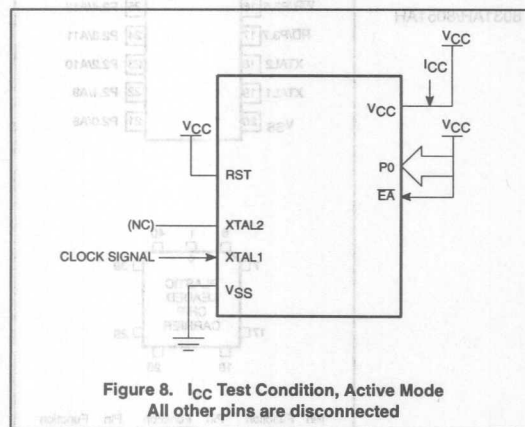
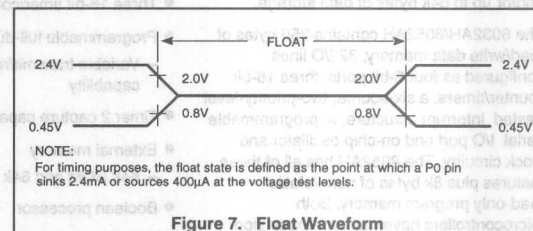
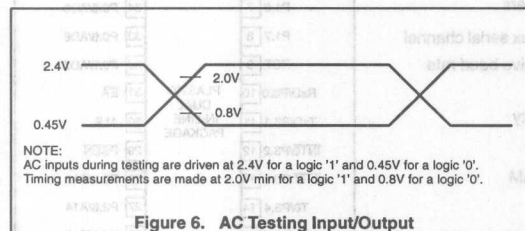
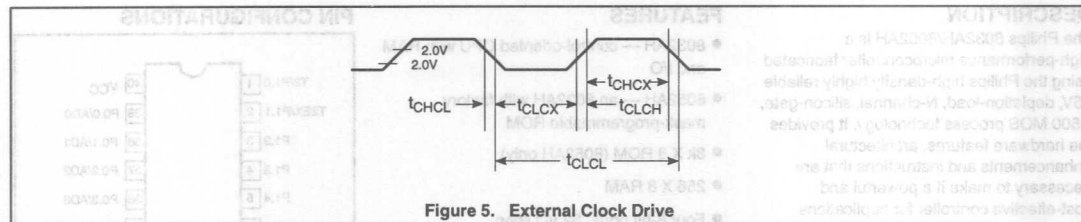


Figure 4. Shift Register Mode Timing



## Single-chip 8-bit microcontroller

## 8031AH/8051AH



## Single-chip 8-bit microcontroller

## 8032AH/8052AH

## DESCRIPTION

The Philips 8032AH/8052AH is a high-performance microcontroller fabricated using the Philips high-density highly reliable +5V, depletion-load, N-channel, silicon-gate, N500 MOS process technology. It provides the hardware features, architectural enhancements and instructions that are necessary to make it a powerful and cost-effective controller for applications requiring up to 64k bytes of program memory and/or up to 64k bytes of data storage.

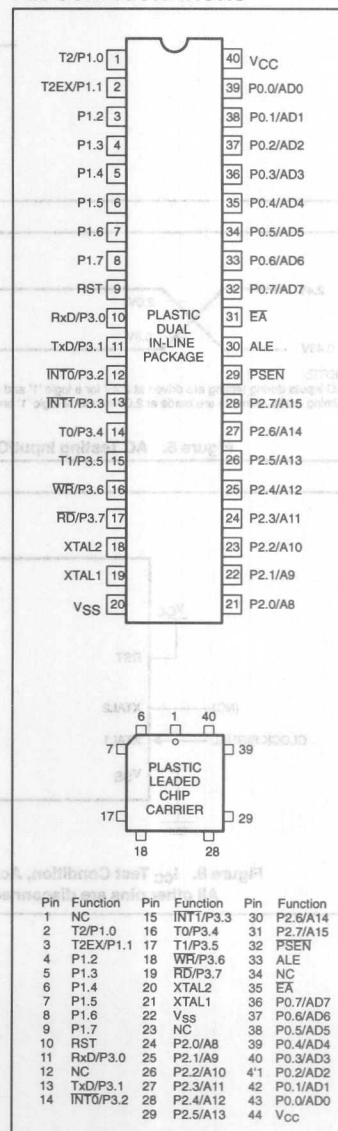
The 8032AH/8052AH contains 256 bytes of read/write data memory, 32 I/O lines configured as four 8-bit ports, three 16-bit counter/timers, a six-source, two-priority-level nested interrupt structure, a programmable serial I/O port and on-chip oscillator and clock circuitry. The 8052AH has all of these features plus 8k bytes of non-volatile read-only program memory. Both microcontrollers have memory expansion capabilities of up to 64k bytes of data storage and 64k bytes of program memory that can be attained with standard TTL compatible memories.

Because of its extensive BCD/binary arithmetic and bit-handling facilities, the 8032AH/8052AH microcontroller is efficient at both computational and control-oriented tasks. Efficient use of program memory is also achieved by using the familiar compact instruction set of the 8031AH/8051AH. Forty-four percent of the instructions are one-byte, 41% two-byte, and 15% three-byte instructions. With a 12MHz crystal, the majority of the instructions execute in just 1.0µs. The longest instructions, multiply and divide, require only 4µs at 12MHz.

## FEATURES

- 8032AH — control-oriented CPU with RAM and I/O
- 8052AH — an 8032AH with factory mask-programmable ROM
- 8k X 8 ROM (8052AH only)
- 256 X 8 RAM
- Four 8-bit ports, 32 I/O lines
- Three 16-bit timer/counters
- Programmable full-duplex serial channel
  - Variable transmit/receive baud rate capability
- Timer 2 capture capability
- External memory
  - 64k ROM and 64k RAM
- Boolean processor
- 128 user bit-addressable locations
- Upward compatible with 8031AH/8051AH

## PIN CONFIGURATIONS



## Single-chip 8-bit microcontroller

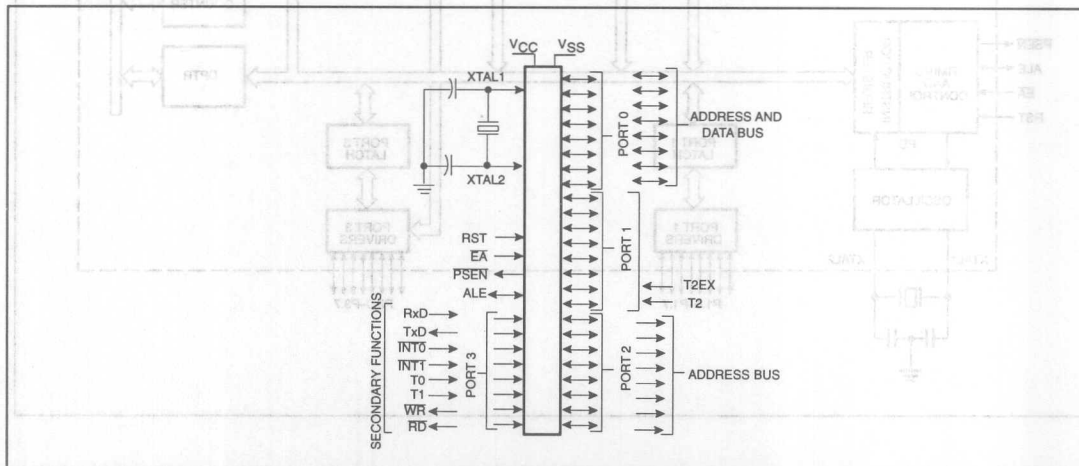
8032AH/8052AH

## ORDERING INFORMATION

BLOCK DIAGRAM

PHILIPS				PHILIPS NORTH AMERICA			TEMPERATURE °C AND PACKAGE	FREQ. MHz
ROMless (MARKING NUMBER)	ROMless (ORDER NUMBER)	ROM	Drawing Number	ROMless	ROM	Drawing Number		
MAB8032AH-2-12P	MAB8032A-2N	MAB8052AH-2P	SOT129	SCN8032HCCN40	SCN8052HCCN40	0415C	0 to +70, Plastic Dual In-line Package	12
MAB8032AH-2-12WP	MAB8032A-2A	MAB8052AH-2WP	SOT187	SCN8032HCCA44	SCN8052HCCA44	0403G	0 to +70, Plastic Leaded Chip Carrier	12
MAF8032AH-2-12P	MAF8032A-2N	MAF8052AH-2P	SOT129	SCN8032HACN40	SCN8052HACN40	0415C	-40 to +85, Plastic Dual In-Line Package	12
MAF8032AH-2-12WP	MAF8032A-2A	MAF8052AH-2WP	SOT187	SCN8032HACA44	SCN8052HACA44	0403G	-40 to +85, Plastic Leaded Chip Carrier	12
MAB8032AH-2-15P	MAB8032A-3N			SCN8032HCFN40	SCN8052HCFN40	0415C	0 to +70, Plastic Dual In-Line Package	15
MAB8032AH-2-15WP	MAB8032A-3A			SCN8032HCFA44	SCN8052HCFA44	0403G	0 to +70, Plastic Leaded Chip Carrier	15
MAF8032AH-2-15P	MAF8032A-3N			SCN8032HAFN40	SCN8052HAFN40	0415C	-40 to +85, Plastic Dual In-Line Package	15
MAF8032AH-2-15WP	MAF8032A-3A			SCN8032HAFA44	SCN8052HAFA44	0403G	-40 to +85, Plastic Leaded Chip Carrier	15

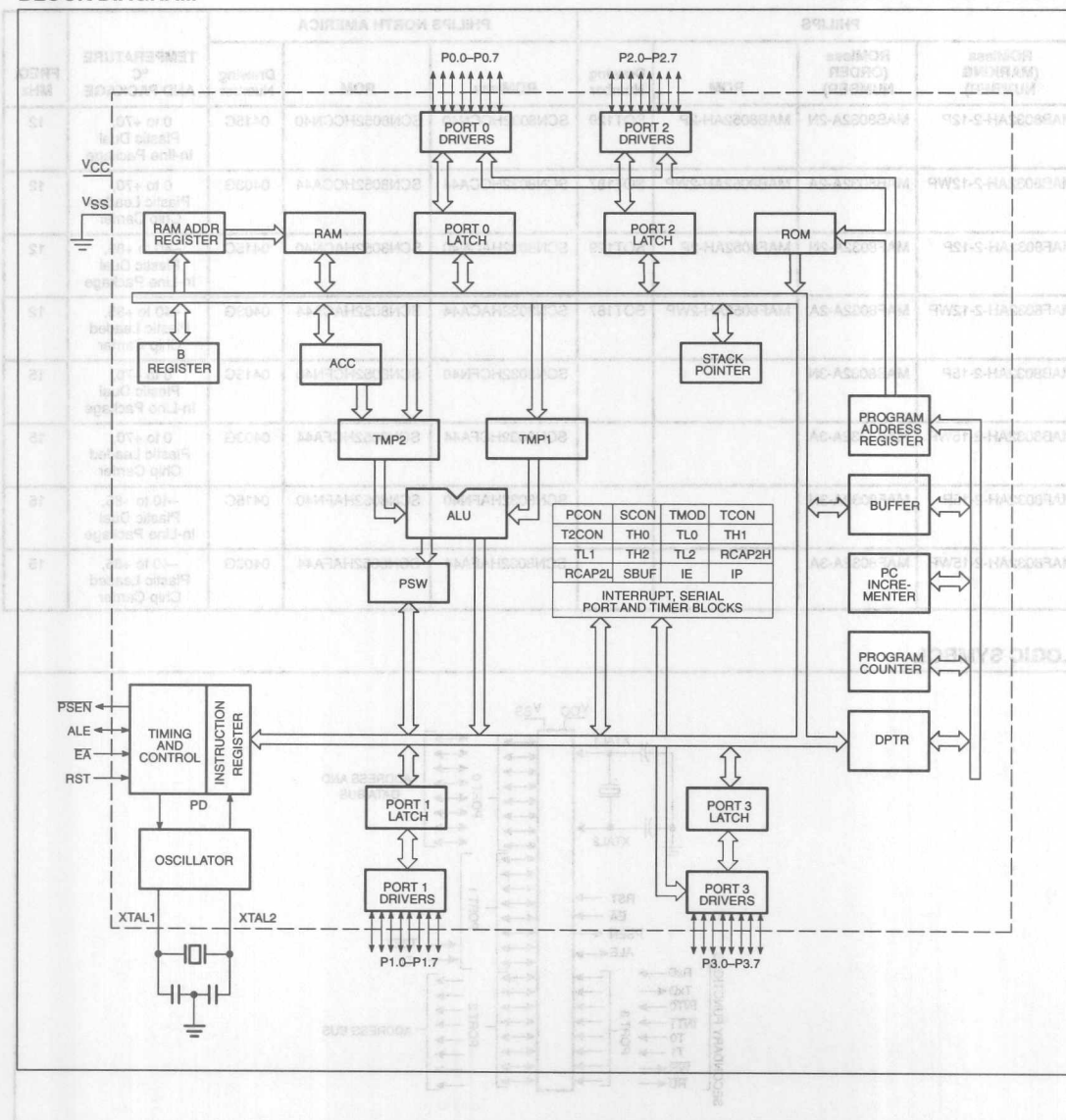
## LOGIC SYMBOL



## Single-chip 8-bit microcontroller

8032AH/8052AH

## BLOCK DIAGRAM



## Single-chip 8-bit microcontroller

8032AH/8052AH

## PIN DESCRIPTION

MNEMONIC	PIN NO.		TYPE	NAME AND FUNCTION
	DIP	LCC		
V <sub>SS</sub>	20	22	I	<b>Ground:</b> 0V reference.
V <sub>CC</sub>	40	44	I	<b>Power Supply:</b> This is the power supply voltage for normal, idle, and power-down operation.
P0.0–0.7	39–32	43–36	I/O	<b>Port 0:</b> Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s.
P1.0–P1.7	1–8	2–9	I/O	<b>Port 1:</b> Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. Pins P1.0 and P1.1 also correspond to the special functions T2, timer 2 counter trigger input, and T2EX, external input to timer 2. the output latch on these two special functions must be programmed to one for that function to operate. Port 1 also receives the low-order address byte during program verification.
	1	2	I	<b>T2 (P1.0):</b> Timer/counter 2 trigger input.
	2	3	I	<b>T2EX (P1.1):</b> Timer/counter 2 external count input.
P2.0–P2.7	21–28	24–31	I/O	<b>Port 2:</b> Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register.
P3.0–P3.7	10–17	11, 13–19	I/O	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 also serves the special features of the SC80C51 family, as listed below:
	10	11	I	<b>RxD (P3.0):</b> Serial input port
	11	13	O	<b>TxD (P3.1):</b> Serial output port
	12	14	I	<b>INT0 (P3.2):</b> External interrupt
	13	15	I	<b>INT1 (P3.3):</b> External interrupt
	14	16	I	<b>T0 (P3.4):</b> Timer 0 external input
	15	17	I	<b>T1 (P3.5):</b> Timer 1 external input
	16	18	O	<b>WR (P3.6):</b> External data memory write strobe
	17	19	O	<b>RD (P3.7):</b> External data memory read strobe
RST	9	10	I	<b>Reset:</b> A high on this pin for two machine cycles while the oscillator is running, resets the device. A small external pull-down resistor (approx 8.2kohm) from RST to V <sub>SS</sub> permits power-on reset when a capacitor (approx 10uF) is also connected from this pin to V <sub>CC</sub> .
ALE	30	33	I/O	<b>Address Latch Enable:</b> Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory.
PSEN	29	32	O	<b>Program Store Enable:</b> The read strobe to external program memory. When the device is executing code from the external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory. PSEN is not activated during fetches from internal program memory.
EA	31	35	I	<b>External Access Enable:</b> EA must be externally held low to enable the device to fetch code from external program memory locations 0000H and 1FFFH. If EA is held high, the device executes from internal program memory unless the program counter contains an address greater than 1FFFH.
XTAL1	19	21	O	<b>Crystal 1:</b> Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	20	O	<b>Crystal 2:</b> Output from the inverting oscillator amplifier.

## OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator, as shown in the logic symbol.

To drive the device from an external clock source, XTAL2 should be driven while XTAL1

is connected to ground. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

## RESET

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running.



## Single-chip 8-bit microcontroller

## 8032AH/8052AH

ABSOLUTE MAXIMUM RATINGS<sup>1, 2, 3</sup>

PARAMETER	RATING	UNIT
Storage temperature range	-65 to +150	°C
Voltage on any other pin to V <sub>SS</sub>	-0.5 to +7.0	V
Input, output current on any single pin	10	mA
Power dissipation	1.5	W

## DC ELECTRICAL CHARACTERISTICS

T<sub>amb</sub> = 0°C to +70°C or -40°C to +85°C, V<sub>CC</sub> = 5V ±10%, V<sub>SS</sub> = 0V<sup>4, 5</sup>

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			MIN	MAX	
V <sub>IL</sub>	Input low voltage		-0.5	0.8	V
V <sub>IH</sub>	Input high voltage; except XTAL2, RST		2.0	V <sub>CC</sub> +0.5	V
V <sub>IH1</sub>	Input high voltage to RST for reset, XTAL2	XTAL1 to V <sub>SS</sub>	2.5	V <sub>CC</sub> +0.5	V
V <sub>OL</sub>	Output low voltage; ports 1, 2, 3 <sup>6</sup>	I <sub>OL</sub> = 1.6mA		0.45	V
V <sub>OL1</sub>	Output low voltage; port 0, ALE, PSEN <sup>6</sup>	I <sub>OL</sub> = 3.2mA		0.45	V
V <sub>OH</sub>	Output high voltage; ports 1, 2, 3	I <sub>OH</sub> = -80µA	2.4		V
V <sub>OH1</sub>	Output high voltage; port 0 in external bus mode, ALE, PSEN <sup>3</sup>	I <sub>OH</sub> = -400µA	2.4		V
I <sub>IL</sub>	Logical 0 input current; ports 1, 2, 3	V <sub>IN</sub> = 0.45V		-800	µA
I <sub>IH1</sub>	Input high current to RST for reset	V <sub>IN</sub> = V <sub>CC</sub> - 1.5V		500	µA
I <sub>I</sub>	Input leakage current; port 0, EA	0.45 < V <sub>IN</sub> < V <sub>CC</sub>		±10	µA
I <sub>IL2</sub>	Logical 0 input current for XTAL2	XTAL1 = V <sub>SS</sub> , V <sub>IN</sub> = 0.45V		-3.2	mA
I <sub>CC</sub>	Power supply current	All outputs disconnected and EA = V <sub>CC</sub>		175	mA
C <sub>IO</sub>	Pin capacitance	f <sub>C</sub> = 1MHz, T <sub>amb</sub> = 25°C		10	pF

T<sub>amb</sub> = -40°C to +85°C - Extended temperature range, V<sub>CC</sub> = 5V ±10%, V<sub>SS</sub> = 0V

V <sub>IH</sub>	Input high voltage; except XTAL2, RST		2.2	V <sub>CC</sub> +0.5	V
V <sub>IH1</sub>	Input high voltage to RST for reset, XTAL2	XTAL1 to V <sub>SS</sub>	2.7		V
I <sub>IL2</sub>	Logical 0 input current for XTAL2	XTAL1 = V <sub>SS</sub> , V <sub>in</sub> = 0.45V		-3.5	mA

## NOTES:

- Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
- For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
- This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
- Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V<sub>SS</sub> unless otherwise noted.
- All voltage measurements are referenced to ground. For testing, all input signals swing between 0.45V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and at output voltages of 0.8V and 2.0V as appropriate.
- V<sub>OL</sub> is derated when the device rapidly discharges external capacitance. This AC noise is most pronounced during emission of address data. When using external memory, locate the latch or buffer as close as possible to the device.

Datum	Emittig	Degraded	
Address	Ports	I/O Lines	V <sub>OL</sub> (Peak Max)
Write Data	P2, P0	P1, P3	0.8V
	P0	P1, p3, ALE	0.8V
7. C <sub>L</sub> = 100pF for port 0, ALE and PSEN outputs; C <sub>L</sub> = 80pF for all other ports.			

## Single-chip 8-bit microcontroller

8032AH/8052AH

## AC ELECTRICAL CHARACTERISTICS

T<sub>amb</sub> = 0°C to +70°C or -40°C to +85°C, V<sub>CC</sub> = 5V ±10%, V<sub>SS</sub> = 0V<sup>1,2</sup>

SYMBOL	FIGURE	PARAMETER	12MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
1/t <sub>CLCL</sub>		Oscillator frequency: <b>Speed Versions</b> SCN8052 C 32MAB8052/32 -2 SCN8052 F 32MAF8052/32 -2			3.5 3.5 3.5 3.5	12 12 15 12	MHz MHz MHz MHz
t <sub>LHLL</sub>	1	ALE pulse width	127		2t <sub>CLCL</sub> -40		ns
t <sub>AVLL</sub>	1	Address valid to ALE low	43		t <sub>CLCL</sub> -40		ns
t <sub>LLAX</sub>	1	Address hold after ALE low	48		t <sub>CLCL</sub> -35		ns
t <sub>LLIV</sub>	1	ALE low to valid instruction in		233		4t <sub>CLCL</sub> -100	ns
t <sub>LLPL</sub>	1	ALE low to PSEN low	58		t <sub>CLCL</sub> -25		ns
t <sub>PLPH</sub>	1	PSEN pulse width	215		3t <sub>CLCL</sub> -35		ns
t <sub>PLIV</sub>	1	PSEN low to valid instruction in		125		3t <sub>CLCL</sub> -125	ns
t <sub>PXIX</sub>	1	Input instruction hold after PSEN	0		0		ns
t <sub>PXIZ</sub>	1	Input instruction float after PSEN		63		t <sub>CLCL</sub> -20	ns
t <sub>AVIV</sub>	1	Address to valid instruction in		302		5t <sub>CLCL</sub> -115	ns
t <sub>PLAZ</sub>	1	PSEN low to address float		20		20	ns
t <sub>PXAV</sub>	1	PSEN to address valid	75		t <sub>CLCL</sub> -8		ns
<b>Data Memory</b>							
t <sub>RLRH</sub>	2, 3	RD pulse width	400		6t <sub>CLCL</sub> -100		ns
t <sub>WLWH</sub>	2, 3	WR pulse width	400		6t <sub>CLCL</sub> -100		ns
t <sub>RLDV</sub>	2, 3	RD low to valid data in		252		5t <sub>CLCL</sub> -165	ns
t <sub>RHDX</sub>	2, 3	Data hold after RD	0		0		ns
t <sub>RHDZ</sub>	2, 3	Data float after RD		97		2t <sub>CLCL</sub> -70	ns
t <sub>LLDV</sub>	2, 3	ALE low to valid data in		517		8t <sub>CLCL</sub> -150	ns
t <sub>AVDV</sub>	2, 3	Address to valid data in		585		9t <sub>CLCL</sub> -165	ns
t <sub>LLWL</sub>	2, 3	ALE low to RD or WR low	200	300	3t <sub>CLCL</sub> -50	3t <sub>CLCL</sub> +50	ns
t <sub>AVWL</sub>	2, 3	Address valid to WR low or RD low	203		4t <sub>CLCL</sub> -130		ns
t <sub>QVWX</sub>	2, 3	Data valid to WR transition	23		t <sub>CLCL</sub> -60		ns
t <sub>QVWH</sub>	2, 3	Data valid to WR high	433		7t <sub>CLCL</sub> -150		ns
t <sub>WHQX</sub>	2, 3	Data hold after WR	33		t <sub>CLCL</sub> -50		ns
t <sub>RLAZ</sub>	2, 3	RD low to address float		20		20	ns
t <sub>WHLH</sub>	2, 3	RD or WR high to ALE high	43	123	t <sub>CLCL</sub> -40	t <sub>CLCL</sub> +40	ns
<b>External Clock</b>							
t <sub>CHCX</sub>	5	High time	20		20		ns
t <sub>CLCX</sub>	5	Low time	20		20		ns
t <sub>CLCH</sub>	5	Rise time		20		20	ns
t <sub>CHCL</sub>	5	Fall time		20		20	ns

## NOTES:

- Parameters are valid over operating temperature range unless otherwise specified.
- Load capacitance for port 0, ALE, and PSEN = 100pF, load capacitance for all other outputs = 80pF.

## Single-chip 8-bit microcontroller

## 8032AH/8052AH

## EXPLANATION OF THE AC SYMBOLS

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

A – Address  
C – Clock  
D – Input data  
H – Logic level high  
I – Instruction (program memory contents)  
L – Logic level low, or ALE  
P – PSEN

Q – Output data  
R – RD signal  
t – Time  
V – Valid  
W – WR signal  
X – No longer a valid logic level  
Z – Float

Examples:  $t_{AVLL}$  = Time for address valid to ALE low.  
 $t_{LLPL}$  = Time for ALE low to PSEN low.

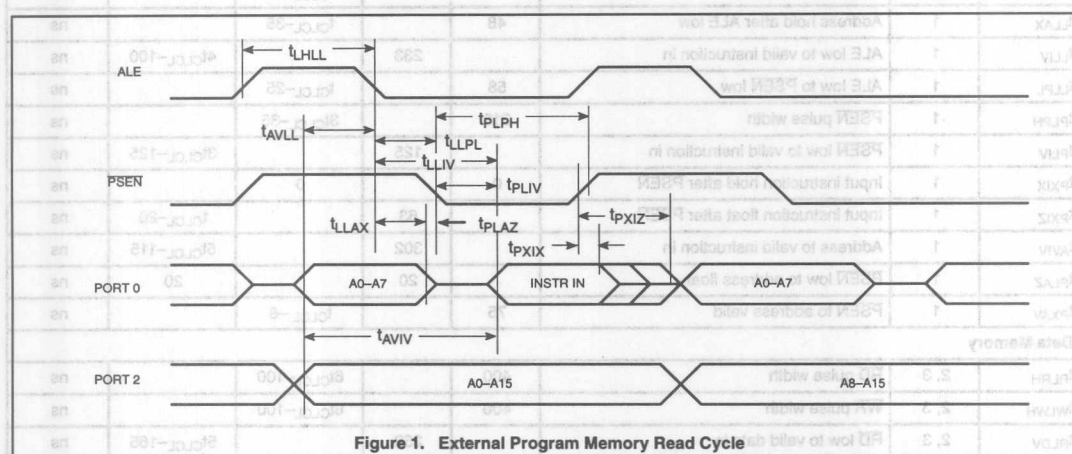


Figure 1. External Program Memory Read Cycle

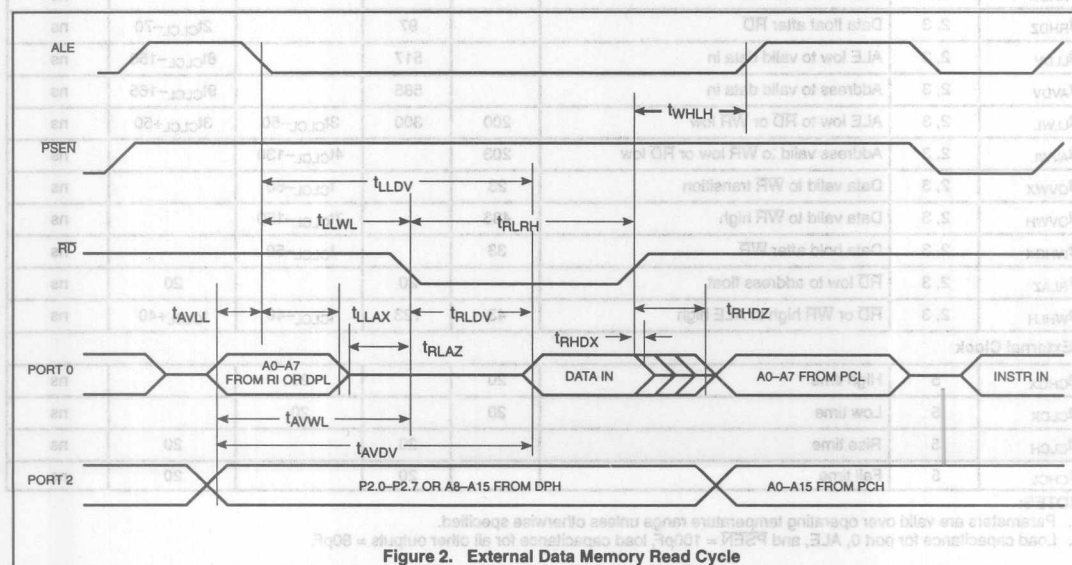
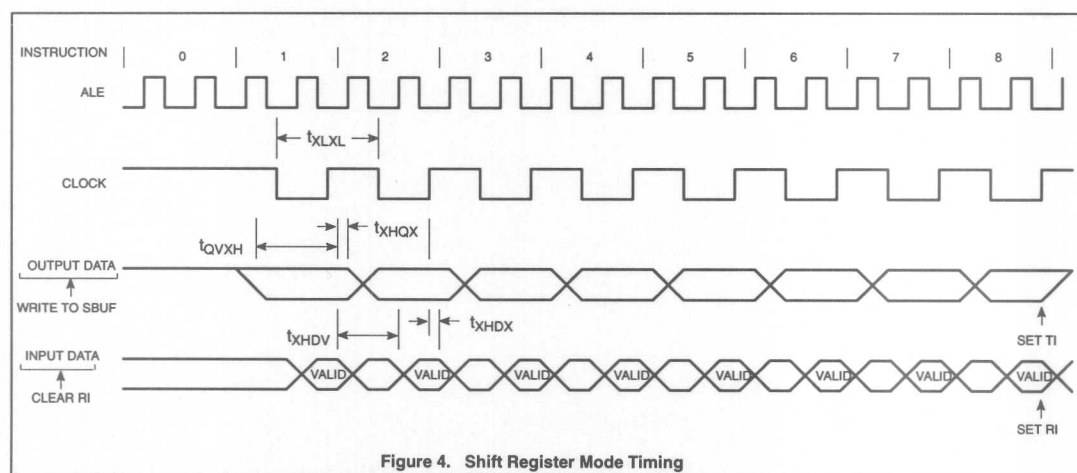
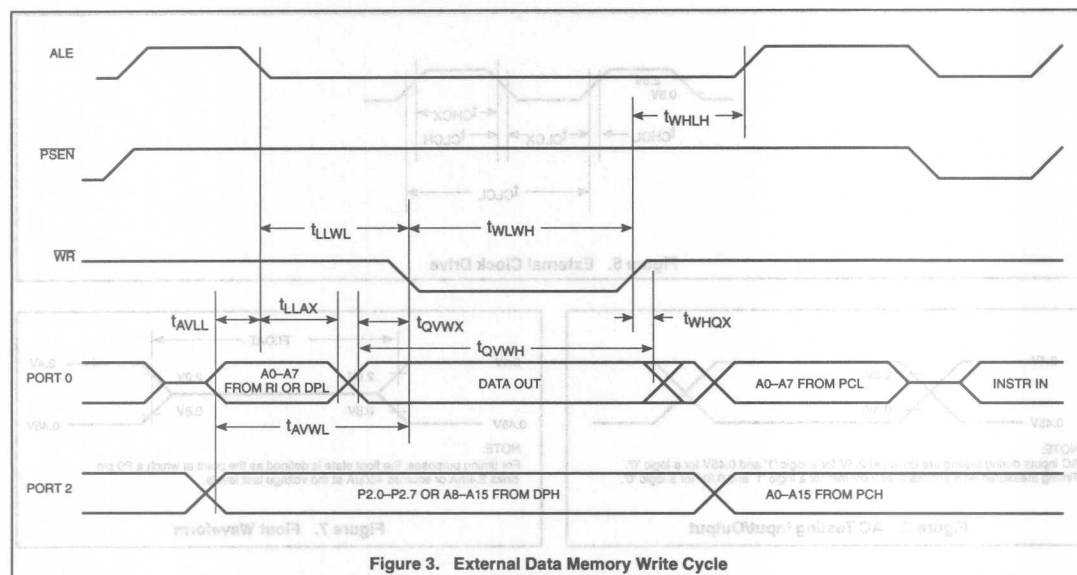


Figure 2. External Data Memory Read Cycle

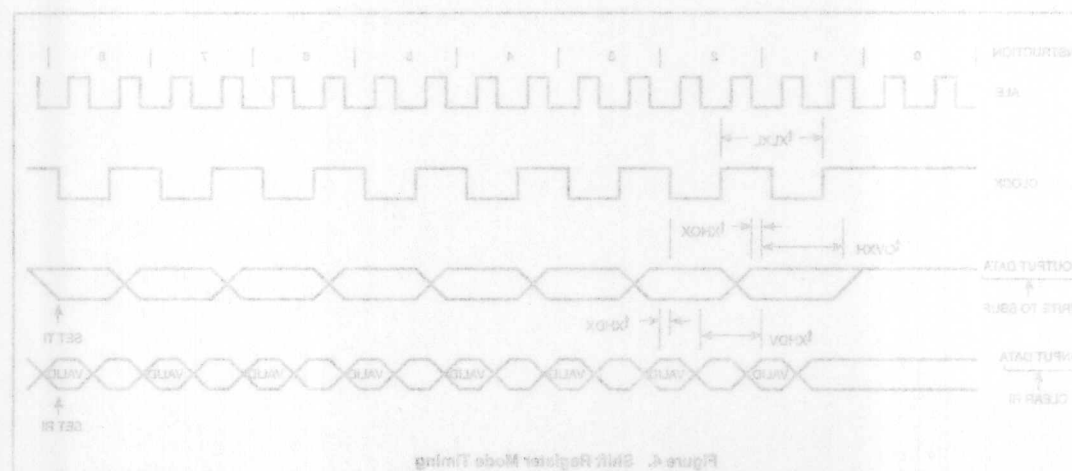
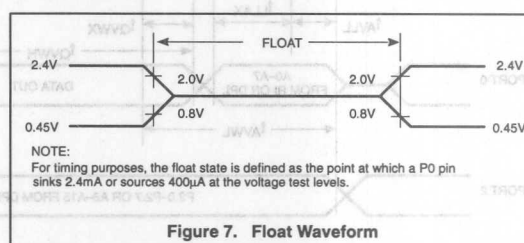
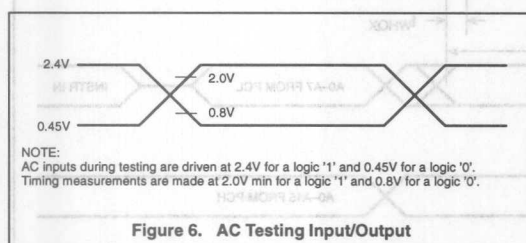
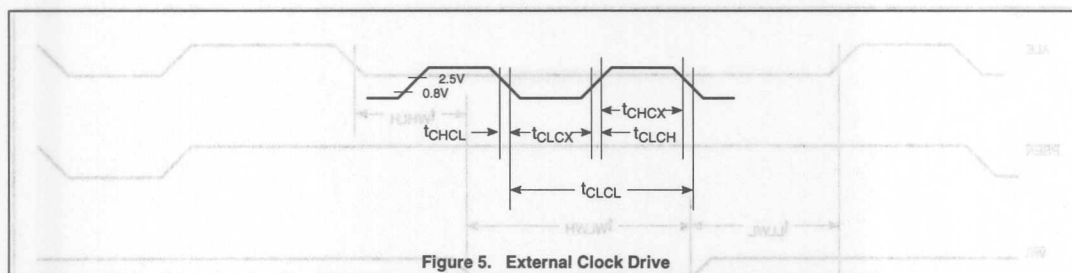
## Single-chip 8-bit microcontroller

8032AH/8052AH



# Single-chip 8-bit microcontroller

8032AH/8052AH





# Section 8

## Development Support Tools

### 80C51-Based 8-Bit Microcontrollers

Philips Semiconductors provides a wide range of development support tools for its 80C51-based 8-bit microcontrollers. As a result, EPROM programming support is available within the programming facilities of many microcontroller development systems.

The following is a list of vendors that offer support for Philips Semiconductors 80C51 microcontroller family.

TELEPHONE	ADDRESS
(523) 83-3344	
(817) 834-7500	
(617) 853-8627	
812-52-82287	
(408) 888-1820	
(603) 858-0137	
31-40-124233	
(805) 371-4008	

TELEPHONE	ADDRESS	COMPANY
(408) 738-2303	1050-L East Oros Ave. Sunnyvale, CA 94086	Active Systems
(800) 528-2105 (713) 461-6430	16821 Washington #190 Houston, TX 77042	BR Microsystems
(202) 581-6144	10525 Wilshire Road N.E. P.O. Box 87045 Redmond, WA 98073-8745	Data NO Corp.
(202) 974-9927	1501 Northwest 68th Place Ft. Lauderdale, FL 33305	Logical Development
(212) 478-0725	P.O. Box 8184 Syosset, NY 11517-0184	Logical Systems
(916) 952-8037	4505 Orange Grove Ave. Sacramento, CA 95841	Needham Electronics
(408) 859-8348	P.O. Box 32698 San Jose, CA 95135	North Valley Products
(801) 736-1006	1008 N. Hob Hill American Fork, UT 84003	Starco Data Communications

### CONTENTS

Development support tools	1276
DB-51 CEIBO Development Board	1285
DS-51 CEIBO In-Circuit Emulator	1287
DS-750 CEIBO Emulator	1295
DS-752 CEIBO Emulator	1297
MP-51 CEIBO Programmer	1299
MetaLink iceMASTER-PE 8051 Family In-Circuit Emulator	1301
MetaLink iceMASTER 8051 Family In-Circuit Emulators	1305
NOHAU EMUL51-PC PC-Based In-Circuit Emulator	1319
LPC 80C51 Family Programmers	1329
PDS51 Development System for 80C51 and Derivatives	1331
S87C00KSD 80C51 - I <sup>2</sup> C Bus Evaluation Board	1335
OM4129 Symbolic Debugging Package XRAY51 for the SDS 8051 Emulator	1336
OM4136 8051 C Cross-Compiler	1339
OM4142 (ASM51) Cross Assembler Package for 80C51/8051-based Systems	1341
OM4144 (PLMT51) PL/M-51 Compiler Package for 80C51/8051-based Systems	1343
SDS 8051 Standalone Debug Station for 80C51/8051-based Systems	1346

### EPROM PROGRAMMING SUPPORT CONTACTS

## Development support tools

### DEVELOPMENT SUPPORT TOOLS

Philips Semiconductors manufactures support tools and also works closely with many "third-party" vendors who provide support tools for our wide variety of 80C51-based microcontroller derivatives.

### Development Systems

In most cases, development systems are available in two versions for ROM and ROMless applications. The ROM emulation products are capable of supporting all versions of a given device type, including EPROM, ROM, and ROMless devices. In contrast, a ROMless emulator can only support applications designed for a ROMless microcontroller. Most development systems are designed to connect to an IBM-PC or compatible personal computer.

### EPROM Programming Support

Philips Semiconductors works closely with major suppliers of EPROM programming equipment to support our family of EPROM microcontrollers. As a result, EPROM programming support is available within the programming facilities of many major distributors.

The following is a list of vendors that offer support for Philips Semiconductors 80C51 microcontroller family.

### DEVELOPMENT SYSTEM CONTACTS

COMPANY	ADDRESS	TELEPHONE
Ashling Microsystems Limited	Plassey Technological Park Limerick, Ireland	(353) 63-334466
BSO Tasking	128 Technology Center P.O. Box 9164 Waltham, MA 02254-9164	(617) 894-7800
Ceibo Ltd.	105 Gleason Rd. Lexington, MA 02173	(617) 863-9927
	Merkazim Building, Industrial Zone P.O. Box 2106 Herzlia 46120, ISRAEL	972-52-555387
Nohau Corp.	51 E. Campbell Ave. Campbell, CA 95008	(408) 866-1820
MetaLink Corp.	325 E. Elliot Road, Suite 23 Chandler, AZ 85225	(602) 926-0797
Philips Semiconductors	Corporate Centre Building BAE-2 P.O. Box 218 5600 MD Eindhoven The Netherlands	31-40-724223
SIGNUM Systems	171 E. Thousand Oaks Blvd., #202 Thousand Oaks, CA 91360	(805) 371-4608

### EPROM PROGRAMMING SUPPORT CONTACTS

COMPANY	ADDRESS	TELEPHONE
Advin Systems	1050-L East Duane Ave. Sunnyvale, CA 94086	(408) 736-2503
BP Microsystems	10681 Haddington #190 Houston, TX 77043	(800) 225-2102 (713) 461-9430
Data I/O Corp.	10525 Willows Road N.E. P.O. Box 97046 Redmond, WA 98073-9746	(206) 881-6444
Logical Devices, Inc.	1201 Northwest 65th Place Ft. Lauderdale, FL 33309	(305) 974-0967
Logical Systems	P. O. Box 6184 Syracuse, NY 13217-6184	(315) 478-0722
Needham's Electronics	4535 Orange Grove Ave. Sacramento, CA 95841	(916) 924-8037
North Valley Products	P.O. Box 32899 San Jose, CA 95152	(408) 929-5345
Strebor Data Communications	1008 N. Nob Hill American Fork, UT 84003	(801) 756-3605

## Development support tools

## Development support tools

## SOFTWARE SUPPORT CONTACTS

## SOFTWARE SUPPORT CONTACTS (Continued)

COMPANY	ADDRESS	TELEPHONE
Franklin Software, Inc.	888 Saratoga Ave. #2 San Jose, CA 95129	(408) 296-8051
Archimedes Software, Inc.	2159 Union St. San Francisco, CA 94123	(415) 567-4010
BSO/Tasking	Tasking Software BV P.O. Box 899 3800 AW Amersfoort The Netherlands  BSO Tasking 128 Technology Center P.O. Box 9164 Waltham, MA 02254-9164	31-33-55-85-84 (Telephone) 31-33-55-00-33 (Fax)  (617) 894-7800 (Telephone) (617) 894-0551 (Fax) (710) 324-0760 (Telex) (800) 458-8276 (Toll Free)

## MICROCONTROLLER DEVELOPMENT SYSTEMS

PRODUCT	DEVICES SUPPORTED
<b>NOHAU CORPORATION</b>	
EMUL51-PC/E32	12MHz Emulator, 32k emulation memory
EMUL51-PC/E128	12MHz Emulator, 128k emulation memory
EMUL51-PC/E32-16	16MHz Emulator, 32k emulation memory
EMUL51-PC/E128-16	16MHz Emulator, 128k emulation memory
EMUL51-PC/E128-20	20MHz Emulator, 128k emulation memory
EMUL51-PC/E128-24	24MHz Emulator, 128k emulation memory
EMUL51-PC/E128-30	30MHz Emulator, 128k emulation memory
EMUL51-PC/E128-33	33MHz Emulator, 128k emulation memory
EMUL51-PC/E128-BSW	12MHz Emulator, 128k bankswitched CODE memory
EMUL51-PC/E128-BSW-16	16MHz Emulator, 128k bankswitched CODE memory
EMUL51-PC/E128-BSW-24	24MHz Emulator, 128k bankswitched CODE memory
EMUL51-PC/E256-BSW	12MHz Emulator, 256k bankswitched CODE memory
EMUL51-PC/E256-BSW-16	16MHz Emulator, 256k bankswitched CODE memory
EMUL51-PC/E256-BSW-24	24MHz Emulator, 256k bankswitched CODE memory
POD-C054	12MHz 83C053, 83C054, 87C054, 87C055 pod
POD-31	12MHz 8031 pod
POD-C31	12MHz 80C31 pod
POD-C31-1	16MHz 80C31 pod
POD-C31-20	20MHz 80C31 pod
POD-C31-24	24MHz 80C31 pod
POD-C31-30	30MHz 80C31 pod
POD-C31-33	33MHz 80C31 pod
POD-32	12MHz 8032 pod
POD-C32	12MHz 80C32 pod
POD-C32-16	16MHz 80C32 pod
POD-C652	12MHz 80C652 pod
POD-C652-16	16MHz 80C652 pod
POD-C51B	12MHz bondout pod for 8051, 80C51, 83C552, 83C652, 83C654, 83C851, and EPROM or ROMless versions of the above
POD-C51B-16	16MHz bondout pod for 8051, 80C51, 83C552, 83C652, 83C654, 83C851, and EPROM or ROMless versions of the above
POD-C51B-24	24MHz version of the above
POD-C52	12MHz 80C32, 80C52, 87C52
POD-C52-16	16MHz 80C32, 80C52, 87C52
POD-CL410	12MHz 80CL31, 80CL51, 83CL410, 83CL610
POD-C451-DIP	12MHz 80C451 DIP pod
POD-C451-DIP-16	16MHz 80C451 DIP pod
POD-C451-PGA	12MHz 80C451 PLCC pod (PGA from pod)
POD-C451-PGA-16	16MHz 80C451 PLCC pod (PGA from pod)
POD-C451B-PGA	12MHz bondout pod for 83C451, 87C451, 80C451 PLCC (PGA from pod)
POD-C451B-PGA-16	16MHz bondout pod for 83C451, 87C451, 80C451 PLCC (PGA from pod)
POD-C528	12MHz 83C524, 87C524, 83C528, 87C528
POD-C528-16	16MHz 83C524, 87C524, 83C528, 87C528

## Development support tools

## Development support tools

## MICROCONTROLLER DEVELOPMENT SYSTEMS (Continued)

## SOFTWARE SUPPORT CONTACTS

PRODUCT	DEVICES SUPPORTED	COMPANY
NOHAU CORPORATION (Continued)		
POD-C550-PGA POD-C550-PGA-16	12MHz 80C550, 83C550, 87C550 16MHz 80C550, 83C550, 87C550	
POD-C552-PGA POD-C552B-PGA	12MHz 80C552 PLCC (PGA from pod) 12MHz bondout pod for 83C552, 87C552, 80C552 PLCC (PGA from pod), 80C562, 83C562	
POD-C552B-PGA-16 POD-C552B-PGA-24 POD-C575 POD-C558-16 POD-CL580	16MHz bondout pod for 83C552, 87C552, 80C552 PLCC (PGA from pod), 80C562, 83C562 24MHz pod for 83552, 87C552, 80C552 12MHz 87C575 16MHz 83CE558, 89CE558 pod 12MHz bondout POD for 83CL580	
POD-C592-PGA POD-C592-PGA-16	12MHz 80C592, 83C592, 87C592 pod 16MHz 80C592, 83C592, 87C592 pod	
POD-C652B POD-C851B	Order as POD-C51B Order as POD-C51B	
POD-C751 POD-C751-16 POD-C752 POD-C752-16 POD-CL782	12MHz 83C750, 87C750, 83C751, 87C751 pod 16MHz 83C750, 87C750, 83C751, 87C751 pod 12MHz 83C752, 87C752 pod 16MHz 83C752, 87C752 pod 12MHz 83CL781, 83CL782, 83CL52 pod	
EMUL51-PC/TR4 EMUL51-PC/TR16 EMUL51-PC/TR4-16 EMUL51-PC/TR16-16	12MHz 4k trace buffer option 12MHz 16k trace buffer option 16MHz 4k trace buffer option 16MHz 16k trace buffer option	
EMUL51-PC/TR16-20 EMUL51-PC/TR16-24 EMUL51-PC/TR16-30 EMUL51-PC/TR16-33	20MHz 16k trace buffer option 24MHz 16k trace buffer option 30MHz 16k trace buffer option 33MHz 16k trace buffer option	
EMUL51-PC/ATR64-16 EMUL51-PC/ATR256-16 EMUL51-PC/ATR64-24 EMUL51-PC/ATR256-24 EMUL51-PC/ATR64-33 EMUL51-PC/ATR256-30	16MHz 64k Advanced Trace Option 16MHz 256k Advanced Trace Option 24MHz 64k Advanced Trace Option 24MHz 256k Advanced Trace Option 33MHz 64k Advanced Trace Option 30MHz 256k Advanced Trace Option	
EMUL51-PC/BOX-CS EMUL51-PC/BOX-CS-20 EMUL51-PC/BOX-CS-24 EMUL51-PC/BOX-CS-30	Serial box with emulator (E128-16) and trace (TR16-16) Serial box with emulator (E128-20) and trace (TR16-20) Serial box with emulator (E128-24) and trace (TR16-24) Serial box with emulator (E128-30) and trace (TR16-30)	
EMUL51-PC/BOX-S	Box with serial port and cable. Box allows operation of emulator external to PC.	
METALINK CORPORATION		
IM-8051/200-20 IM-8051/400-20	iceMASTER-8051 emulator Model 200, 32K emulation memory, 20MHz iceMASTER-8051 emulator Model 400, 32K emulation memory, 4K trace buffer, 2 performance analyzers, 20MHz	
IM-8051/400-24	iceMASTER-8051 emulator Model 400, 128K emulation memory, 4K trace buffer, 2 performance analyzers, 24MHz	
128KUP 752/1 PGMPIC	128K memory expansion option for iceMASTER-8051 Programmer accessory for iceMASTER to program 87C751, 87C752	
8031-12PC 8031-16PC 8031-20PC 8031-24PC	0.5 to 12MHz 8031, 80C31 0.5 to 16MHz 8031, 80C31 0.5 to 20MHz 8031, 80C31 0.5 to 24MHz 8031, 80C31	
8032-12PC 8032-16PC 8032-20PC	0.5 to 12MHz 8031, 80C31, 8032, 80C32 0.5 to 16MHz 8031, 80C31, 8032, 80C32 0.5 to 20MHz 8031, 80C31, 8032, 80C32	

## Development support tools

## Development support tools

## MICROCONTROLLER DEVELOPMENT SYSTEMS (Continued)

PRODUCT	DEVICES SUPPORTED
<b>METALINK CORPORATION (Continued)</b>	
8032-24PC	0.5 to 24MHz 8032, 80C32, 8031, 80C31
8052-12PC	0.5 to 12MHz 8031, 80C31, 8032, 80C32, 8051, 8751, 80C51, 87C51, 8052, 8752, 80C52, 87C52
8052-16PC	0.5 to 16MHz 8031, 80C31, 8032, 80C32, 8051, 8751, 80C51, 87C51, 8052, 8752, 80C52, 87C52
80410-12PC	0.5 to 12MHz 80CL410
80451-12PC	1.2 to 12MHz 80C451
80451-16PC	1.2 to 16MHz 80C451
80528-12PC	1.2 to 12MHz 80C528
80528-16PC	1.2 to 16MHz 80C528
80552-12PC	1.2 to 12MHz 80C552, 80C562
80552-16PC	1.2 to 16MHz 80C552, 80C562
80652-12PC	1.2 to 12MHz 8031, 80C31, 80C652
80652-16PC	1.2 to 16MHz 8031, 80C31, 80C652
80851-12PC	1.2 to 12MHz 8031, 80C31, 80C851
83053-12PC	6 to 12MHz 83C053, 83C054, 87C054
83451-12PC	1.2 to 12MHz 80C451, 83C451, 87C451
83528-12PC	1.2 to 12MHz 80C528, 83C528, 87C528, 83C524, 87C524
83528-16PC	1.2 to 16MHz 80C528, 83C528, 87C528, 83C524, 87C524
83550-10PC	1.2 to 10MHz 80C550, 83C550, 87C550
83552-12PC	1.2 to 12MHz 80C552, 83C552, 87C552, 80C562, 83C562
83552-16PC	1.2 to 16MHz 80C552, 83C552, 87C552, 80C562, 83C562
83652-12PC	1.2 to 12MHz 80C652, 83C652, 87C652, 80C552, 83C552, 87C552, 80C562, 83C562
83652-16PC	1.2 to 16MHz 80C652, 83C652, 87C652, 80C552, 83C552, 87C552, 80C562, 83C562
83654-12PC	1.2 to 12MHz 80C652, 83C652, 87C652, 83C654, 87C654, 80C552, 83C552, 87C552, 80C562, 83C562
83654-16PC	1.2 to 16MHz 80C652, 83C652, 87C652, 83C654, 87C654, 80C552, 83C552, 87C552, 80C562, 83C562
83751-12PC	0.5 to 12MHz 83C751, 87C751
83751-16PC	0.5 to 16MHz 83C751, 87C751
83752-12PC	0.5 to 12MHz 83C752, 87C752



## Development support tools

## Development support tools

## MICROCONTROLLER DEVELOPMENT SYSTEMS (Continued)

PHILIPS SEMICONDUCTORS				PROBE								
C TYPE					SUPPORTS				ADAPTER	ADAPTER SUPPORTS		
			Probe/ Probe base	Probe Head	DIL	PLCC	QFP	Misc		DIL	PLCC	QFP
Core	C51	C MOS	OM1092									
80C												
8XC	053	C MOS	OM5054	+ OM1097 + OM1097 + OM5012 + OM5012  + OM4111 + OM5055 + OM1095 + OM1095  + OM4112 + OM1096 + OM1096	42							
8XC	054	C MOS	OM5054		42	44				42	44	
80C	31	C MOS	OM1092		40	44			+ OM4117	-	-	44
80C	51	C MOS	OM1092		40	44			+ OM4117	-	-	44
80C	32	C MOS	OM1079		40							
80C	52	C MOS	OM1079		40							
8XC	451	C MOS	OM4123		-	66			+ OM4124	64	-	-
8XC	528	C MOS	OM4110		40	44			+ OM4117	-	-	44
8XC	550	C MOS	OM4110		40	44				40	44	
8XC	552	C MOS	OM1092		-	66			+ OM4118	-	-	50
8XC	562	C MOS	OM1092		-	66			+ OM4118	-	-	50
8XC	575	C MOS										
8XC	592	C MOS	OM4110		-	66						
8XC	652	C MOS	OM1092		40	44			+ OM4117	-	-	44
8XC	654	C MOS	OM1092		40	44			+ OM4117	-	-	44
8XC	750	C MOS										
8XC	751	C MOS	OM1094	524	28			+ OM4117	524	28		
8XC	752	C MOS	OM5072	28	28				28	28		
8XC	851	C MOS	OM1092	40	44			+ OM4117	-	-	44	
8XC	852	C MOS	OM4119	+ OM4118/1 + OM4118/2 + OM4118/3	- - -	- - -	- - -	c_less Iso minor				
80CE												
8XCE	556	C MOS	OM4110	+ OM4271	-	-	-	-	+ OM4115	-	-	60
8XCE	598	C MOS	OM4110	+ OM4114	-	-	-	-	+ OM4115	-	-	60
8XCE	654	C MOS	OM1092	+ OM1096	-	-	-	-	+ OM4115	-	-	44
80CL												
80CL	31	SAC CMOS	OM1079	+ OM5012 + OM5012 + OM4840 + OM4840  + OM 5004 + OM5012 + OM5012	40							
80CL	51	SAC CMOS	OM1079		40							
80CL	32	SAC CMOS	OM1079		40							
80CL	52	SAC MOS	OM1079		40							
8XCL	167/267	SAC MOS	OM1079		S 64					S 64		
8XCL	168/268	SAC MOS	OM1079		S 64					S 64		
8XCL	410	SAC MOS	OM1079		40					40		
8XCL	411	SAC MOS	OM1079		40				64 P			
8XCL	500	SAC MOS	OM1097		-	-	-	-	conn.	-	-	64
8XCL	751	SAC MOS	OM1079		40							
8XCL	752	SAC MOS	OM1079		40							

## Development support tools

PROGRAMMER		ADAPTER SUPPORTS			REMARKS		
Programmer	SUPPORTS			Adapter	ADAPTER SUPPORTS		
	DIL	PLCC	QFP		DIL	PLCC	QFP
<b>80C</b>							
OM4232	42	44	-	OM4236	-	-	44
OM4232	40	44	-	OM4236	-	-	44
OM4231	-	66	-	-	-	-	80
OM4232	40	44	-	OM4236	-	-	44
OM4231	-	66	-	OMXXXX	-	-	80
OM4232	-	-	-	OM4235	-	-	65
OM4232	40	44	-	OM4236	-	-	44
OM4232	40	44	-	OM4236	-	-	44
OM4231	XX	XX	-	-	-	-	80
OM4231	XX	XX	-	-	-	-	80
OM4231	XX	XX	-	-	-	-	80
<b>80CE</b>							
(OM4232 7)	-	-	-	OMXXXX	-	-	80
	-	-	44	OM4237	-	-	80
<b>80CL</b>							
							Attention: no erasable versions but "piggybacks" (P85CL000, P85CL580, P85CL781)

## Development support tools

## Development support tools

## EPROM MICROCOMPUTER PROGRAMMING SUPPORT

DEVICE	MANUFACTURER/MODEL	MODULE/ADAPTOR	SOFTWARE VERSION
87C054 SDIP	N. Valley Products Philips, Ceibo MP-51	SAM-054SD PPA-054SD	
87C51 DIP	Advin Sailor-PAL/SA, /SB BP Microsystems EP-1140 Ceibo MP-51 (PMP51SD) Data I/O Unisite 40 Data I/O Unipak 2b Data I/O Series 1000 Logical Devices ALLPRO N. Valley Products SPGM-100 Philips LCPX5X40 (P8051LCP40) Strebtor PLP-S1A	Adaptor-8751  PPA-51XSD  351B103 SR40  SAM-51SD  MC4851DIP	V2.2 V16 V05 (Use Intel 87C51 menu) V1.47 V1.0
87C51 PLCC	Ceibo MP-51 (PMP51SD) Data I/O Unisite 40 Data I/O Unipak 2b Logical Devices ALLPRO N. Valley Products SPGM-100 Philips LCPX5X40 (P8051LCP40)	PPA-51XSD Chipsite 351B103P Required SAM-51ASD	V2.3 V16 V1.47 V1.0
87C52 DIP	BP Microsystems EP-1140 Ceibo MP-51 (PMP51SD) Data I/O 29B, Unipak 2b Data I/O Unisite 40 N. Valley Products SPGM-100 Philips LCPX5X40 (P8051LCP40)	PPA-XSD 351B103  SAM-52SD	V23 V3.1
87C451 DIP	Ceibo MP-51 (PMP51SD) Logical Devices ALLPRO N. Valley Products SPGM-100	PPA-451SD PPRequired SAM-451SD	V1.47 V1.0
87C451 PLCC	Advin Sailor-PAL/SA, /SB Ceibo MP-51 (PMP51SD) N. Valley Products SPGM-100 Data I/O Unisite Philips LCPX5X (P8051LCPX)	Adaptor-87451 PPA-451ASD SAM-451ASD Chipsite	V1.0 V2.8
87C528 DIP	BP Microsystems EP-1140 Ceibo MP-51 N. Valley Products SPGM-100 Philips LCPX5X40 (P8051LCP40)	PPA-XSD SAM-528	
87C528 PLCC	Ceibo MP-51 (PMP51SD) N. Valley Products SPGM-100 Philips LCPX5X40 (P8051LCP40)	PPA-51XSD SAM-528A	
87C552	Ceibo MP-51 (PMP51SD) N. Valley Products SPGM-100 Data I/O Unisite Philips LCPX5X (P8051LCPX)	PPA-552ASD SAM-552ASD Chipsite	V2.2 V3.1
87C652 DIP	BP Microsystems EP-1140 Ceibo MP-51 N. Valley Products Philips LCPX5X40 (P8051LCP40)	PPA-51XSD SAM-52SD	
87C652 PLCC	Ceibo MP-51 (PMP51SD) Philips LCPX5X40 (P8051LCP40)	PPA-51XSD	
87C654 DIP	BP Microsystems EP-1140 Ceibo MP-51 (PMP51SD) N. Valley Products SPGM-100 Philips LCPX5X40 (P8051LCP40)	PPA-51XSD SAM-654SD	
87C654 PLCC	Ceibo MP-51 (PMP51SD) Philips LCPX5X40 (P8051LCP40)	PPA-51XSD	

Development support tools

Development support tools

EPROM MICROCOMPUTER PROGRAMMING SUPPORT (Continued)

DEVICE	MANUFACTURER/MODEL	MODULE/ADAPTOR	SOFTWARE VERSION
87C751 DIP	Advin Sailor-PAL/SA, /SB	EM-751	
	BP Microsystems EP-1140	HEAD-40A	
87C751 PLCC	Ceibo MP-51 (PMP51SD)	PPA-751SD	
	Data I/O Unisite 40		
87C752 DIP	Data I/O Unisite 40	351B113D	
	Logical Devices ALLPRO	OPTAPC-751	
87C752 PLCC	N. Valley Products SPGM-100	SAM-751SD	
	Needham's Electronics		
87C751 PLCC	MetaLink	752/1 PGMPC	
	Logical Systems (Sunshine EW-901)	PA751	
87C751 PLCC	Philips LCPX5X (P8051LCPX)		
	Strebor PLP-S1A	MC7512DIP	
87C751 PLCC	Ceibo MP-51 (PMP51SD)	PPA-51XSD	
	Data I/O Unisite 40	Chipsite	
87C752 DIP	Logical Devices ALLPRO	Required	
	N. Valley Products SPGM-100	SAM-751ASD	
87C752 DIP	Advin Sailor-PAL/SA, /SB	EM-751	
	BP Microsystems EP-1140	HEAD-40A	
87C751 PLCC	Ceibo MP-51 (PMP51SD)	PPA-752SD	
	Data I/O Unisite 40		
87C752 PLCC	N. Valley Products SPGM-100	SAM-752SD	
	Needham's Electronics		
87C751 PLCC	MetaLink	752/1 PGMPC	
	Logical Systems (Sunshine EW-901)	PA751	
87C751 PLCC	Philips LCPX5X (P8051LCPX)	OPTAPC-752	
	Strebor PLP-S1A	MC7512DIP	
87C751 PLCC	Ceibo MP-51 (PMP51SD)	PPA-752ASD	
	Data I/O Unisite 40	Chipsite	
87C752 PLCC	N. Valley Products SPGM-100	SAM-752ASD	

NOTE:  
Philips programmers are available in the U.S.A. through Philips Semiconductors distributors.

## Development support tools

## ADDITIONAL PROGRAMMING SUPPORT

DEVICE	MANUFACTURER	MODULE/ADAPTOR	COMMENTS
87C51/52/652/ 654/528 PLCC	Logical Systems	PA51-44	Use with any 40-pin microcontroller programming site that supports the appropriate EPROM size.
87C51/52/652/ 654/528 QFP		PA52-QFP	Use with any 40-pin microcontroller programming site that supports the appropriate EPROM size.
87C451 DIP		PA451-64	Use with any 87C51 40-pin programming site.
87C451 PLCC		PA451-68	Use with any 87C51 40-pin programming site.
87C550 DIP		550BASE	Use with any 87C51 40-pin programming site.
87C550 PLCC		PA550-44	Use with any 87C51 40-pin programming site.
87C552 PLCC		PA552-68	Use with any 8752/C52/C252/C51FA 40-pin programming site.
87C752 PLCC		PA28-28	Use with any 87C752 28-pin DIP programmer.
87C751 PLCC	Philips	No part number assigned	This adapter allows programming the 87C751 PLCC part in conjunction with any programmer that can already program the DIP version of the part.

## MICROCONTROLLER SUPPORT

PRODUCT	DEVICES SUPPORTED	MANUFACTURER	DESCRIPTION
8051 C Compiler	8051 and derivatives	Franklin Software	C Compiler for 8051 family
8051 C Compiler	8051 and derivatives	Archimedes Software	C Compiler for 8051 family
80C51 C Compiler	8051 and derivatives	BSO/Tasking	C Compiler for 8051 family
P8051DB	8051 and derivatives	Ceibo	80C51 Family Development Board
S87C00KSD	--	Philips	I <sup>2</sup> C Demonstration Board. 87C751 controls various I <sup>2</sup> C peripherals. Board has sockets for 87C752, 87C652, and 87C552 also.
--	--	Philips	The Philips computer Bulletin Board system has available a microcontroller newsletter, application and demonstration programs for download, and the ability to send messages to microcontroller applications engineers. Access by modem at 2400, 1200, or 300 baud. The telephone numbers are: (800) 451-6644 (in the U.S.) or (408) 991-2406.
SMI-CNV451SD	80/83/87C451	Philips	Philips product adapts a PLCC emulator plug for the 80C451 to the DIP pinout.

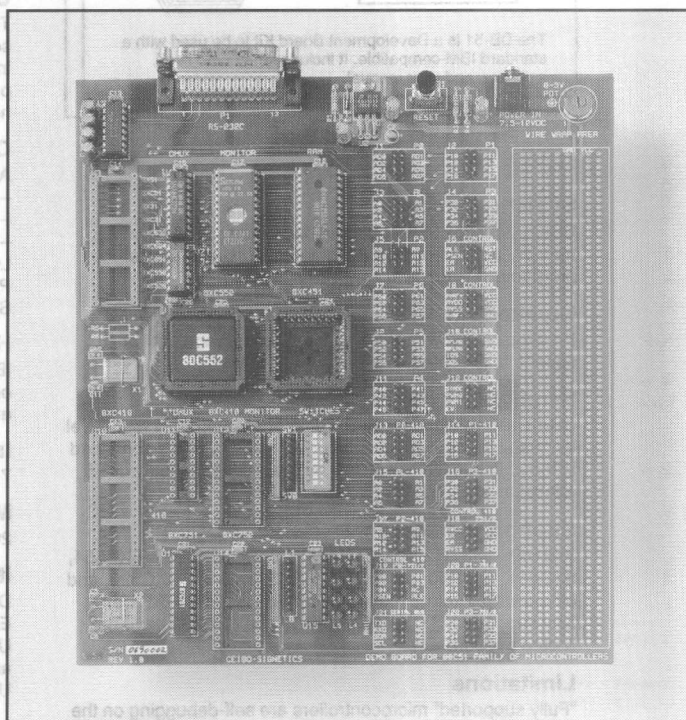


# DB-51

## CEIBO Development Board

DB-51 is a high-performance system design board dedicated to the Philips 80C51 family of microcontrollers. It provides an easy-to-use flexible instrument which enables the user to build a primary prototype, analyze and debug it, make changes, and continue debugging. And you can improve your design decisions by using the DB-51 to check and test the advantages of several different microcontrollers. The DB-51 is also a great training and tutorial aid for becoming familiar with designs using the 80C51 architecture. Note that the DB-51 is not intended to replace a full emulator system in complex microcontroller designs.

- Supports most of the Philips 80C51 derivative microcontrollers
- Serially linked to IBM PC or compatible hosts
- 32K of user code memory
- Software breakpoints
- Examine and alter chip registers, RAM, and ports
- Symbolic debugger compatible with linker object files
- Source Level Debugger for Assembler, PLM and C.
- Software Trace and Performance Analyzer
- Upload and download of object and hex files
- Special wire-wrap area for prototyping
- User's manual with examples and applications designed to familiarize the user with 8XC51 architecture and programming as well as the use of the DB-51 itself



DB-51 is a high-performance system design board dedicated to the Philips 80C51 family of microcontrollers. It provides an easy-to-use flexible instrument which enables the user to build a primary prototype, analyze and debug it, make changes, and continue debugging. And you can improve your design decisions by using the DB-51 to check and test the advantages of several different microcontrollers. The DB-51 is also a great training and tutorial aid for becoming familiar with designs using the 80C51 architecture. Note that the DB-51 is not intended to replace a full emulator system in complex microcontroller designs.

ORDERING INFORMATION  
Order part number P081D8D from your  
local Philips Semiconductor Distributor.

# SPECIFICATIONS



## System Memory

DB-51 provides 32K of user code memory. This RAM memory permits downloading and modifying of users' programs.

## Breakpoints

Breakpoints allow real-time program execution until an opcode is executed at a specified address.

## Symbolic Debugger

DB-51 allows symbolic debugging of assembler or high-level languages. The symbolic debugger uses symbols contained in the absolute file generated by the most commonly used relocater and linker programs.

## Supported Microcontrollers

8XC31/51, 8XC32/52, 8XC31/51, 8XC32/52, 8XC652, 8XC654, 8XC851, 8XC550, 8XC552, 8XC562, 8XC451, 8XC528, and others with external memory addressing and a UART are fully supported. 8XCL410, 8XC751 and 8XC752 have very limited support.

## Limitations

"Fully supported" microcontrollers are self-debugging on the DB-51. Thus, some of the chip resources are used by the board: the monitor program uses the bottom 32K of program memory; chips are always operated in the external memory mode; the UART is used to communicate to the PC and is thus not normally available to the user program; interrupt response is slowed slightly by re-vectoring from the monitor

program to the user program; use of watchdog timers and power-down and idle modes of operation are limited due to interaction with the monitor program.

"Limited support" microcontrollers do not have on-chip UARTs and most do not support external program memory. Thus, download of programs to these parts is not supported on the DB-51. The 87C751 provided with the board is pre-programmed with a "micro" monitor program and some predefined experiments described in the user manual. Also, these parts use the I<sup>2</sup>C bus to communicate to the PC, limiting the use of I<sup>2</sup>C for other purposes.

## User Software

The board is provided with a very easy-to-use menu-driven software program as well as command oriented user interface software. On-line assembler and disassembler are provided together with upload and download capabilities of hexadecimal and object files.

## Command Set

ASM – BIT – BYTE – BREAKPOINT [enable, disable, reset]  
 – CHIP [type] – CLS – CODE – DATA – DASM – DEFAULT  
 – DIR – EVALUATE – EXIT – GO [from, till] – HALT – HELP  
 – HISTORY – LINES – LIST [file] – LOAD [code, symbols] –  
 LOCALS – MODULES – PORTS – PROCEDURES –  
 PUBLICS – RBIT – RBYTE – REGISTERS – RESET –  
 SAVE – SOUND – STATUS – STEP [n] – TIME

## Host Characteristics

IBM PC/XT/AT or compatible system with 512 kbyte of RAM, one floppy disk drive, one RS-232 interface board for the PC and cable, PC-DOS 2.0 or later.

## Input Power

7.5 VDC to 12.0 VDC (9 VDC wall transformer supplied).

## Mechanical Dimensions

20 cm × 25 cm

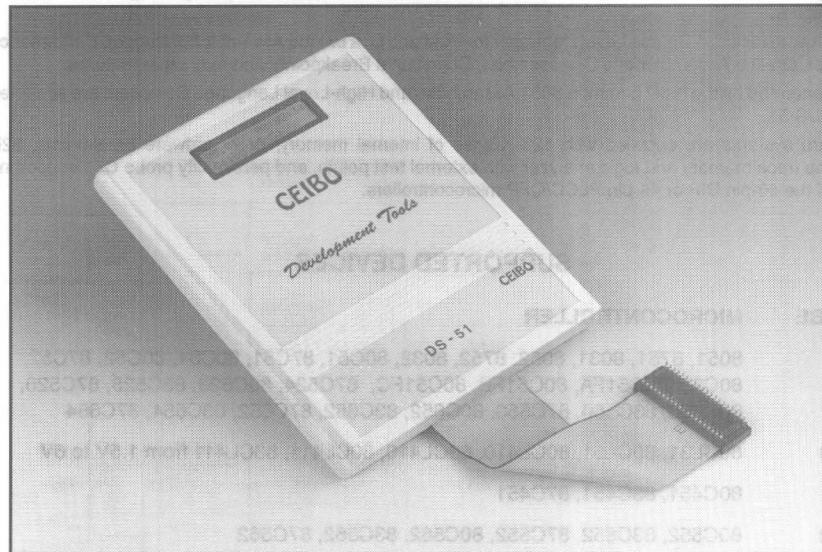
## Items Supplied as Standard

DB-51 board, 80C552 and 87C751 microcontrollers, monitor EPROM, power supply, RS-232 cable.  
 User software including symbolic debugger, on-line assembler and disassembler.  
 User's manual and operating instructions.

## ORDERING INFORMATION

Order part number P8051DBSD from your local Philips Semiconductors Distributor.

# DS-51 CEIBO In-Circuit Emulator



*In-Circuit Emulator for 8051 Family of Microcontrollers*

## FEATURES

- Real-time and transparent in-circuit emulator
- Supports most of the 8051 derivatives
- Emulates 1.5V to 6V microcontrollers
- Maximum frequency of 42MHz
- 128K of internal memory
- 32K trace memory and logic analyzer
- 64K hardware and conditional breakpoints
- Source Level Debug for Assembler, PLM and C
- Serially linked to IBM PC at 115 KBaud

CEIBO TEL: 314-830-4084 FAX: 314-830-4083

## DS-51 CEIBO In-Circuit Emulator

### DESCRIPTION

DS-51 is a real-time in-circuit emulator dedicated to the 8051 family off microcontroller. It is serially linked to a PC/XT/At or compatible host and carries out a transparent emulation on the target microcontroller.

The system emulates almost every 8051 derivative in the complete voltage and frequency range specified by the microcontroller manufacturer.

DS-51 also supports the new low-power and low-voltage 8051 microcontrollers and derivatives and can emulate the microcontrollers using either the built-in 5V power supply or any voltage applied to the target circuitry. This selection is done by means of software control. The permitted voltage range is from 1.5V to 6V or higher.

The software includes Source Level Debugger for PLM and C, a unique Assembler Debugger, Performance analyzer, On-line Assembler and Disassembler, Conditional Breakpoints and may other features.

Files generated by the most common 8051 Assemblers and High-Level Language Compilers are accepted by the DS-51.

Standard systems are supplied with 128 KBytes of internal memory, 64K hardware breakpoints, 32K real-time trace memory and logic analyzer with external test points, and personality probe C51 supporting most of the 40-pin DIP or 44-pin PLCC/QFP microcontrollers.

### SUPPORTED DEVICES

PROBE	MICROCONTROLLER
C51	8051, 8751, 8031, 8052, 8752, 8032, 80C51, 87C51, 80C31, 80C52, 87C52, 80C32, 80C51FA, 80C51FB, 80C51FC, 87C524, 80C528, 83C528, 87C528, 80C550, 83C550, 87C550, 80C652, 83C652, 87C652, 83C654, 87C654
C410	80CL31, 80CL51, 80CL410, 83CL410, 80CL411, 83CL411 from 1.5V to 6V
C451	80C451, 83C451, 87C451
C552	80C552, 83C552, 87C552, 80C562, 83C562, 87C562
C558	80C558, 83C558, 89C558
C575	80C575, 83C575, 87C575, 80C576, 83C576, 87C576
C580	80CL580, 83CL580 from 1.5V to 6V
C592	80C592, 83C592, 87C592
C598	80C598, 83C598, 87C598
C752	83C750, 87C750, 83C751, 87C751, 83C752, 87C752
C781	80CL781, 83CL781, 80CL782, 83CL782 from 1.5V to 6V
C851	80C851, 83C851

Consult Ceibo for other microcontrollers support.

CEIBO TEL: 314-830-4084 FAX: 314-830-4083

**SPECIFICATIONS****Emulation Memory**

DS-51 provides 128 KBytes of code memory with software mapping and banking capabilities.

**Hardware Breakpoints**

Up to 65536 hardware breakpoints allow real-time program execution until an opcode is executed at a specified address line of your source code, data contents and read/write operation.

**Conditional Breakpoints**

A complete set of conditional breakpoints permit halting program emulation on code addresses, source code lines, access to external data, and on-chip memory, port and register contents.

**Performance Analyzer**

This useful function checks the software trace buffer and provides time statistics on modules and procedures as a percentage of the total execution time.

**Languages and File Formats**

DS-51 accepts files with Intel OMF51 object or hex format. Other assemblers and high-level languages such as C with OMF51 format (Franklin, Archimedes, IAR, MCC, BSO/Tasking, etc.) are also supported.

**Source Level Debugger**

The DS-51 Software includes a high-level language Source Level Debugger. This program may be used to debug code written Assembler, PLM and C. The Source Level Debugger includes commands which allow the user to get all the information needed for testing the programs and hardware in real-time. The commands permit setting breakpoints on high-level language lines, adding a watch window with the symbols and variables of interest, modifying variables, displaying floating point values, showing the trace buffer, executing assembly steps and many more useful functions.

**Personality Probes**

DS-51 uses standard and bond-out microcontrollers for hardware and software emulation. The selection of a different microcontroller is made by replacing the microcontroller on the probe or changing the probe. The personality probes run at a frequency of the crystal on them or from the clock source supplied by the user hardware. Therefore, the same probe may be adapted to you frequency requirements. The minimum and maximum frequencies are determined by the emulated chip characteristics, while the emulator maximum frequency for standard probes is 42MHz.

**Trace and Logic Analyzer**

The trace memory is 32 KByte deep and is used to record the microprocessor activities. Eight lines are user selectable test points. Trigger inputs and conditions are available for starting and stopping the trace recording. The trace buffer can be viewed in disassembled symbolics, frames or high level language source code.

**Host Characteristics**

IBM PC/XT/AT or compatible systems with 640 KBytes of RAM, one hard disk, one RS-232C interface card for the PC, PC-DOS 5.0 or later.

**Items Supplied As Standard**

In-circuit emulator with 64 KByte breakpoints, 128 KByte Internal Code Memory. Personality probe C51 for 8051 microcontrollers. User software including source level debugger, on-line assembler and disassembler. User's Manual and Operating Instructions. RS-232 Interface Cable. Power Supply and Power Cord.

**Options**

Personality probes for the different microcontrollers. Adapter for 44-pin PLCC devices.



## TRACE CAPABILITIES

## External Trace Start/Stop Triggers

DS-51 has two External Trigger signals that allow starting and stopping of the trace recording upon external events.

## Stop Trace when Full or Continuous Recoding Mode

There are two trace recoding modes: Cyclic and Trace Full. In the Cyclic Mode the trace is continuously filled with recorded data. In the Full Mode recoding stops when trace is full.

## Selectable Trace Trigger Levels

The trigger state permits selecting the way trigger signals behave. The active mode may be either level or edge for the external start and stop trigger signals.

## Trace Status on the Fly

Trace Status allows viewing information without stopping emulation. The Trace Status includes buffer full, buffer empty, length, etc.

## Trace Filtering on Address Ranges

Up to 10 different ranges are allowed to filter the recorded data into the trace memory. The start and stop addresses of modules and procedures can be entered using special prefixes.

TRACE DUMP			
Inst#	Address	Instruction	
(0028h)	01C7h	ret	
40 3		REG_P1 = PATTERN_CPL;	
(0029h)	0164h	mov REG_P1,PATTERN_CPL	
41 3		CALL CPL_P1(8);	
(002Ah)	0167h	mov ROT_NO,#08h	
(002Bh)	016Ah	acall CPL_P1 (0146).	
31 3		DO CPL_CNT = 1 TO CPL_NO;	
(002Ch)		OT_CNT,#01h	
(002Dh)	Goto...	,ROT_CNT	
(002Eh)	Origin	c	
(002Fh)		a,ROT_NO	
(0030h)	Mode SOURCE	TESTS#0035 (015F)	
32 3		DELAY;	
(0031h)	0150h	acall DELAY (01A4)	
(0032h)	01A4h	mov 09h,#01h	
(0033h)	01A7h	mov DELAY_CNT,#00h	
(0034h)	01AAh	mov r6,DELAY_VALUE	
(0035h)	01ACh	mov r7,09h	
(0036h)	01AEh	mov r4,DELAY_VALUE	
(0037h)	01B0h	mov r5,0Bh	
(0038h)	01B2h	mov a,r5	
(0039h)	01B3h	clr c	
Goto a specific trace point			

## BREAKPOINT OPTIONS

## Breakpoint on Data Read/Write and Address

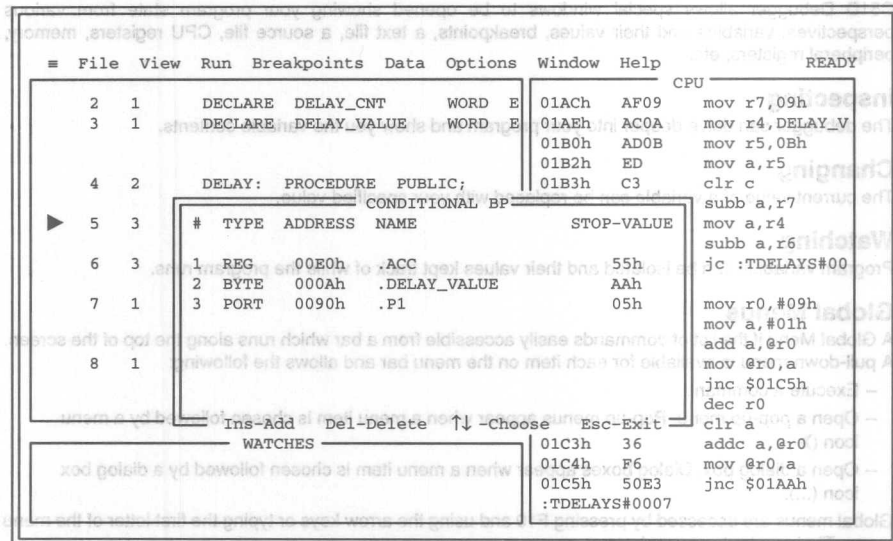
Real-time Hardware Breakpoints can be set on Data Read, Data Write, Data Read or Write and Opcode Fetch in any address range.

## Stop Emulation on Pass Count

Pass Count on any event including opcode may be specified by the number of occurrences.

## Breakpoint on External Events

Trace Testpoints may be used as external Breakpoints on Hardware events if enabled. The External Breakpoint triggers may also be the Trace Full condition, then program execution will stop once it is full. The Logic Operator commands allow selection of AND/OR combinations of the two external breakpoints. The options are both low, any one is low, both are high, only one is high, both are leading edge, etc.



## ABOUT THE DEBUGGER

### Environment

C51D is a menu-driven program supplied with DS-51. This debugger runs either under DOS or Windows. C51D Debugger is used to load a program, execute it in real-time or simulate the software environment and many other functions.

### Tracing

A program may be executed one line at a time. You can trace a program using high-level language lines or assembly instructions.

### Stepping

This is like tracing but program execution steps over CALL instructions without leaving the current procedure.

### Viewing

C51D Debugger allows special windows to be opened showing your program state from various perspectives: variables and their values, breakpoints, a text file, a source file, CPU registers, memory, peripheral registers, etc.

### Inspecting

The debugger can delve deeper into your program and show you the variable contents.

### Changing

The current value of a variable can be replaced with your specified value.

### Watching

Program variables can be isolated and their values kept track of while the program runs.

### Global Menus

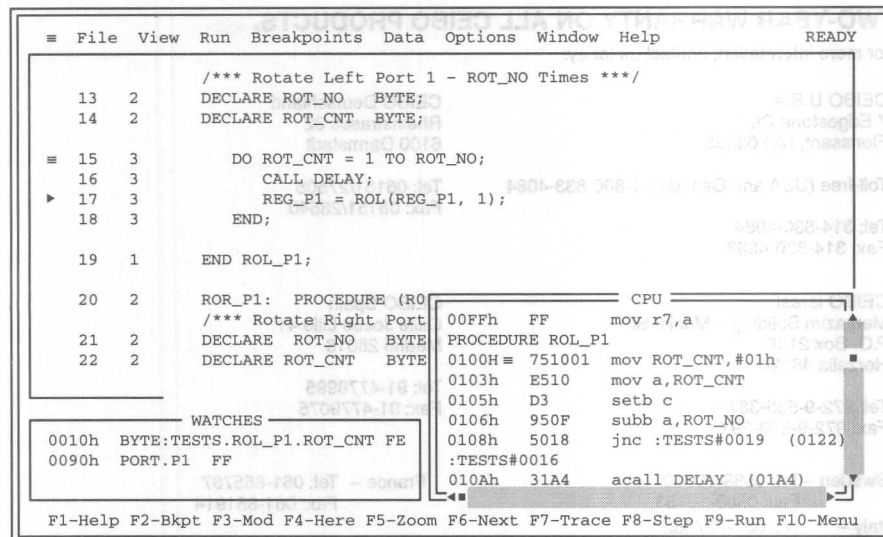
A Global Menu if the list of commands easily accessible from a bar which runs along the top of the screen. A pull-down menu is available for each item on the menu bar and allows the following:

- Execute a command
- Open a pop-up menu. Pop-up menus appear when a menu item is chosen followed by a menu icon (➤).
- Open a dialog box. Dialog boxes appear when a menu item is chosen followed by a dialog box icon (...).

Global menus are accessed by pressing F10 and using the arrow keys or typing the first letter of the menu name. Their contents never change.

Some of the menu commands have hot key shortcuts that are available from any part of the C51D Debugger.

## ABOUT THE DEBUGGER



## Local Menus

C51D Debugger is context-sensitive and uses Local Menus specifying different windows. Local menus are tailored to the particular window you are in. It is important not to confuse them with global menus. To prompt a local menu press Alt-F10 or click the right button of your mouse. Menu placement and contents depends on which window or pane you are in and where your cursor is. Contents may vary from one local menu to another. Many local commands appear in almost all local menus. The results of these similarly-named commands may differ, depending on the context. Every command on a local menu has a hot key shortcut consisting of Ctrl plus the highlighted letter in the command. Because of this setup, a hot key, like Ctrl-S might mean one thing in one context but something quite different in another. The core commands are still consistent across local menus. For example, the Goto command and the Search command always do the same thing, even when they are invoked from different windows.

## Input Boxes

Many of the C51D Debugger command options are available in input boxes. An input box prompts you to type in a string, i.e., the name of a file.

## Windows

C51D Debugger displays all information and data in both global and local menus, dialog boxes (where options are set and information entered) and windows. There are many window types depending on the kind of information it holds. Windows may be opened and closed using menu commands (or hot key shortcuts for those commands). Most of the Debugger windows come from the View menu. After a window has been opened, you can move, resize, close, and otherwise manage them with commands from the Window and System menus (=).

## WARRANTY

## TWO-YEAR WARRANTY ON ALL CEIBO PRODUCTS.

For more information, contact us today:

CEIBO U.S.A.  
7 Edgestone Ct.  
Florissant, MO 63033

Toll-free (USA and Canada): 1-800-833-4084

Tel: 314-830-4084  
Fax: 314-830-4083

CEIBO Deutschland  
Rheinstrasse 32  
6100 Darmstadt

Tel: 06151/27505  
Fax: 06151/28540

CEIBO Israel  
Merkazim Building – Maskit St.  
P.O. Box 2106  
Herzlia 46120

Tel: 972-9-555-387  
Fax: 972-9-553-297

CEIBO Spain  
Calle Josue Lillo 41  
Madrid 28018

Tel: 91-4778995  
Fax: 91-4779075

Sweden – Tel: 0589-10950  
Fax: 0589-16153

Italy – Tel: 051-727252  
Fax: 051-727515

France – Tel: 061-855767  
Fax: 061-851914

Other Countries: Fax: 972-9-553-297

Local Menus  
The CEIBO Debugger is context-sensitive and uses Local Menus specifying different windows. Local menus are allowed in the particular window you are in. It is important not to confuse them with global menus. To prompt a local menu press Alt-F10 or click the right button of your mouse. Menu placement and contents depends on which window or pane you are in and where your cursor is. Contents may vary from one local menu to another. Many local commands appear in almost all local menus. The results of these similarly-named commands may differ, depending on the context. Every command on a local menu has a hot key shortcut consisting of Ctrl plus the highlighted letter in the command. Because of this setup a hot key like Ctrl-S might mean one thing in one context but something quite different in another. The core commands are still consistent across local menus. For example, the Goto command and the Search command always do the same thing, even when they are invoked from different windows.

Input Boxes  
Many of the CEIBO Debugger command options are available in input boxes. A input box prompts you to type in a string, i.e., the name of a file.

Windows  
The CEIBO Debugger displays all information and data in both global and local menus, dialog boxes (where options are set and information entered) and windows. There are many window types dependent on the kind of information it holds. Windows may be opened and closed using menu commands (or hot key shortcuts for those commands). Most of the Debugger windows come from the View menu. After a window has been opened, you can move, resize, close, and otherwise manage them with commands from the Window and System menus (a).

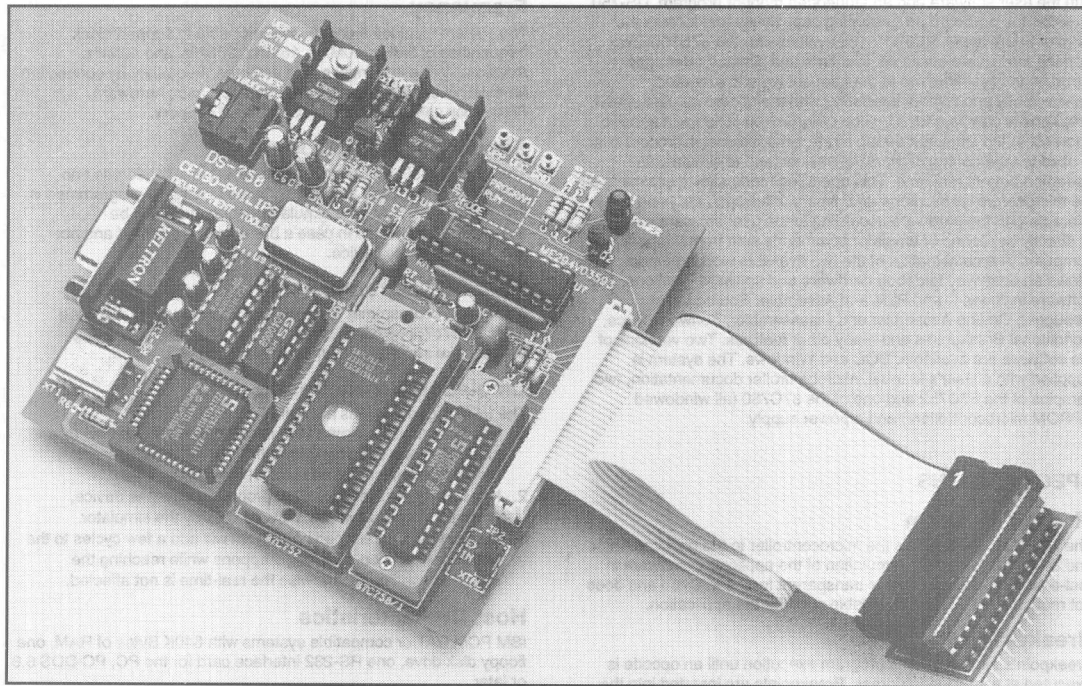
CEIBO TEL: 314-830-4084 FAX: 314-830-4083



# DS-750 Microcontroller

## CEIBO Development Tool

Part Number: P750EM SD



Development Tool for 83/87C750 Microcontrollers

### FEATURES

- Emulates 87C750 and 83C750 microcontrollers in real-time
- Programmable clock up to 40MHz
- Built-in programmer for 87C750/1/2
- Simulator debug mode
- Source-level debugger for C, PLM and Assembler
- DOS and Windows Software
- 24-pin DIP emulation header
- Serially linked to IBM PC at 115K baud

PART NUMBER: P750EM SD  
Available through Philips Semiconductors, authorized distributors.

DS-750 is a development tool that supports Philips 87C750 microcontrollers at any frequency allowed by the devices. It is serially linked to a PC/XT/AT or compatible system and can emulate the microcontroller using either the built-in clock oscillator or any other clock source connected to the microcontroller. The clock oscillator generates 40MHz, 20MHz, 16MHz, 10MHz and 5MHz. Emulation is carried out by programming an 87C752 microcontroller with the user software and an embedded monitor program. DS-750 provides the on-board programming capabilities and locates the monitor in the upper 1K that is not available for the 87C750. Two working modes are available: real-time and simulator debugger mode. In the real-time mode the user software is executed transparently and without interfering with the microcontroller speed. Breakpoints can be added to stop program execution at a specific address. In the simulator debug mode, an additional microcontroller is used to take control of the 87C750 lines and to simulate its operation but not real-time. This operating mode allows access to all the microcontroller functions (I/O, timers, interrupts, etc.) and interacts with the hardware according to the user software execution or directly by means of emulator commands sent from the host computer. The combination of the two available working modes allows an easy way to debug hardware and software functions. The software includes C and PLM and Assembler Source Level Debugger, On-line Assembler and Disassembler, Software Trace, Conditional Breakpoints and many other features. Two versions of the software are available: DOC and Windows. The system is supplied with a User's Manual, microcontroller documentation, two samples of the 87C752 and one of the 87C750 (all windowed EPROM microcontrollers) and a power supply.

## SPECIFICATIONS

### Monitor Program

The Monitor Program links the microcontroller to the host computer and is used to control the emulation of the application software in real-time. The monitor code is transparent for the 87C750 and does not reduce the 1K Byte of available code for the application.

### Breakpoints

Breakpoints allow real-time program execution until an opcode is executed at a specified address. Breakpoints are inserted into the application program and then the Emulation EPROM Device is programmed with the application code. The breakpoints can be disabled by results in adding a few cycles to the program execution.

### User Software

DS-750 program has two software environments: DOS and Windows 3.X. The DOS program is based on pull-down menus. The additional Windows program carries out the same functions of the DOS program with all the features and benefits of the new operating system.

### Symbolic Debugger

DS-750 allows symbolic debugging of assembler or high-level languages. The symbolic debugger uses symbols contained in the absolute file generated by the most commonly used Assemblers and High Level Language Compilers.

### Source Level Debugger

The DS-750 software includes a source level debugger for Assembler and High-Level Languages (PLM, C and others) with the capability of executing lines of the program while displaying the state of any variable.

## Software Trace

Program execution can be recorded in a 64K buffer. Conditional breakpoints may be defined to stop program execution. The user can define events and variables to be added to the software trace. The software trace is not a real-time function and is performed by slowing down the emulation speed.

## Frequency

The system includes a crystal oscillator able to support clock frequencies of 5MHz, 10MHz, 16MHz, 20MHz, and 40MHz. Additionally, the user may select any other frequency by connecting an external clock source through the application hardware. Frequency selection is done by means of jumpers.

## Simulator Debug Mode

The simulator allows breakpoints to be set at any address and condition even though the user software is actually programmed in the 87C752 EPROM. The simulator debug mode will be automatically activated in case a breakpoint is enabled and not programmed in the device.

## Built-in Programmer

The built-in programmer may be used to program the following devices: 87C750, 87C751, and 87C752. All the programming features like security and encryption are fully supported.

## Emulation Restrictions

The following restrictions are valid for DS-750:

1. The system uses some of the microcontroller resources to emulate it: one interrupt (either INTO or INT1 according to software selection) and 5 bytes of the internal stack.
2. If you specify a breakpoint not programmed in the device, program execution will be slowed down by the simulator.
3. Disabling a programmed breakpoint will add a few cycles to the program execution. This only happens while reaching the specified condition, otherwise the real-time is not affected.

## Host Characteristics

IBM PC/XT/AT or compatible systems with 640K Bytes of RAM, one floppy disk drive, one RS-232 interface card for the PC, PC-DOS 5.0 or later.

## Input Power

120 VAC to 15 VDC wall transformer supplied.

## Mechanical Dimensions

10cm x 10cm.

## Items Supplied as Standard

Development Systems with built-in programmer. 24-pin emulation header. User software including source level debugger, on-line assembler and disassembler. User's Manual and Operating Instruction. RS-232 interface cable. Power supply.

## Warranty

Six months limited warranty, parts and labor.

## Ordering Information

**PART NUMBER: P750EM SD**

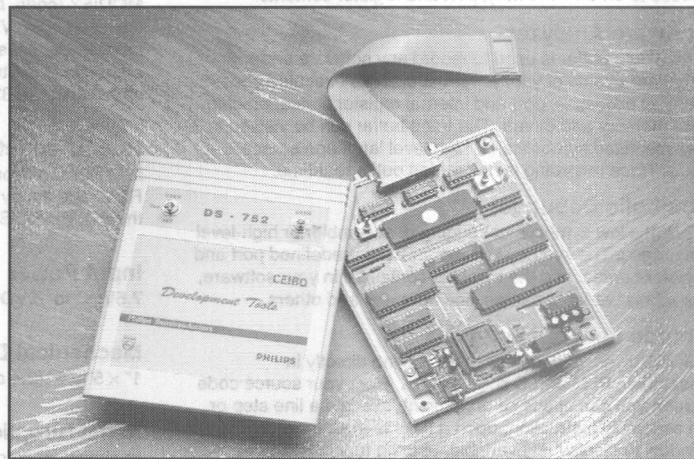
Available through Philips Semiconductors' authorized distributors.

# DS-752

## CEIBO Emulator

DS-752 is a real-time, high performance microcontroller development system dedicated to the 8XC751/8XC752 single-chip microcontrollers. It provides an easy-to-use and flexible instrument which reduces the development and debugging cycle and enables the user to solve hardware and software problems quickly and efficiently. It operates with an IBM PC or compatible computer and carries out complete real-time and transparent emulation of the target CPU.

- Real time and transparent in-circuit emulator
- Supports Philips 83C751/83C752 and 87C751/87C752 microcontrollers
- Symbolic Debugger compatible with Intel object files
- Source Level Debug for C and PLM
- Source Level Assembler Debugger
- 2K hardware breakpoints and conditional breakpoints
- 2K of internal memory
- 64K Software Trace
- Performance Analyzer
- Serially linked to IBM PC or compatible hosts
- On-line Assembler and Disassembler
- Easy to follow pull-down menus and windows



Part No: P752EM SD
Available through Philips distributors.
12MOC 8850-554-10805

PC-DOS, PC/XT/AT are trademarks of IBM.  
IBM is a registered trademark of IBM Corporation.  
Intel is a registered trademark of Intel Corporation.

# SPECIFICATIONS

## Emulation Memory

DS-752 provides 2KBytes of code memory.

## Hardware Breakpoints

Up to 2,048 hardware breakpoints allow real-time program execution until an opcode is executed at a specified address or line of your source code.

## Conditional Breakpoints

A complete set of conditional breakpoints permits halting program emulation on code addresses, source code lines, access to on-chip memory, port and register contents.

## Software Analyzer

A 64 KByte buffer is used to record any software and hardware events of your program, such as executed code, memory accesses, port and internal register states, on-chip data memory and others. The trace buffer can be viewed in disassembled symbolics, or high level language source code. Trace recording is not carried out in real-time.

## Symbolic Debugger

DS-752 allow symbolic debugging of assembler or high-level languages. The symbolic debugger uses predefined port and register names, and the symbols contained in your software, like labels, variable names, line numbers and others.

## Source Level Debugger

DS-752 gives full support for debugging directly in assembler, PLM and C source code. From your source code screen you can specify a breakpoint, execute a line step or an assembly instruction, open a flexible-in-size watch window to display any variable, use the function keys to display the trace memory, registers and data, redefine the Program Counter, and reset the microprocessor.

## Languages and File Formats

DS-752 accepts files generated by Intel software (Assembler, PLM) or compatibles in hex or object format. Other assemblers and high-level languages such as C with Intel compatible format (Franklin, Archimedes, IAR, etc.) are also supported.

## Personality Adapters

DS-752 uses standard microcontrollers for hardware and software emulation. The selection of a different microcontroller is made by software commands and using a supplied socket adapter. The systems run at the frequency of the crystal on them or from the clock source supplied by the user hardware. Therefore, the same system may be adapted to your frequency requirements.

The minimum frequency is determined by the emulated chip characteristics, while maximum frequency for standard probes is 16MHz.

## Supported Devices

83C751, 87C751, 83C752, 87C752.

## Command Set

The available functions include: FILE (load, save), DEBUG (go, halt, reset, source level debugger, breakpoints), MODIFY (code, byte, registers, ports, assembler, disassembler), VIEW (watch window, publics, modules, procedures, lines, symbols, file, dir), ANALYZER (conditional breakpoints, software trace, event watch), SETUP (default, chip, base, rs-232 port).

## Host Characteristics

IBM PC/XT/AT or compatible system with 640 KBytes of RAM, one floppy disk drive, one RS-232C interface card for the PC, PC-DOS 2.0 or later.

## Input Power

7.5VDC to 12VDC or 5VDC/500mA from the user circuit.

## Mechanical Dimensions

1" x 5" x 6" (2.4 cm x 13 cm x 15 cm).

## Items Supplied as Standard

In-circuit emulator with 2 KByte breakpoints, 2 KByte Internal Code Memory. Personality adapter for 24-pin DIP microcontrollers. User Software including source level debugger, on-line assembler and disassembler. User's Manual and Operating Instructions. RS-232 Interface Cable. 9 VDC wall transformer.

## Warranty

Six months limited warranty; parts and labor.

## Ordering Information

<b>Part No: P752EM SD</b>
Available through Philips distributors.
12NC: 9350-554-10602



# MP-51

## CEIBO Programmer

MP-51 is an EPROM, PLD and Microcontroller Programmer dedicated to standard 24 to 32-pin EPROMs, all of the microcontrollers belonging to the 8051 family, and high density PLDs. Its modern design provides a powerful low-cost and high performance instrument, easy to use and conveniently sized.

MP-51 operates with an IBM PC/XT/AT or compatible personal computer, and carries out a set of powerful functions on the selected device. An RS-232 interface is used to link MP-51 to a PC.

The unit consists of the instrument and adaptors. The adaptors may be replaced to suit the user's requirements. Adaptors are available for all the possible packages such as DIP, QFP, LCC and PLCC.

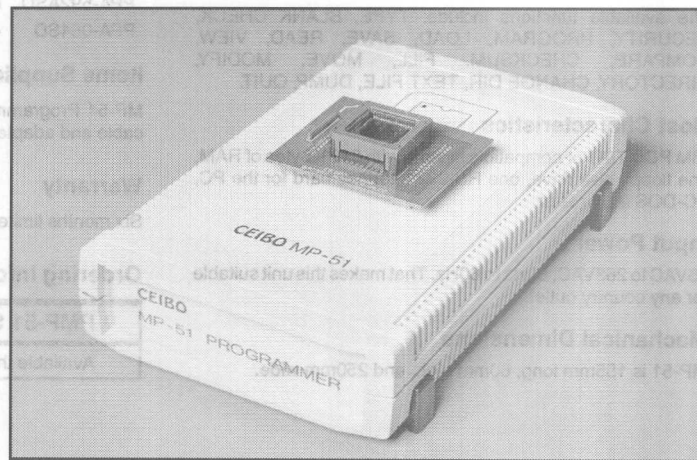
MP-51 software handles a PC Memory Buffer where code is loaded from a disk or filled with the contents of a device. Furthermore, this buffer may be saved on a disk file, parts of the buffer can be moved from one location to another, filled with a constant, or modified by the user. The Memory Buffer can be displayed, and finding values or strings in it is possible.

Before programming, MP-51 checks if the installed adaptor is compatible with the device type selected by the user. This test is done before programming any device.

MP-51 has the capability to check if the device is totally erased, and can also compare if the contents of the plugged device are equal to the contents of the Memory Buffer. Address range can be specified for both operations.

MP-51 allows to enable or disable the PLD or Microcontroller security capabilities and handles the Lock Bit 1, Lock Bit 2, Lock Bit 3 and Encryption Table available in several Microcontrollers.

- High quality EPROM, PLD and Microcontroller Programmer
- Serially linked to IBM PC/XT/AT or other compatible host computers
- Loads/Saves Hex, Binary, Object and JEDEC files to and from disk
- Easy to follow windows and pull-down menus
- Supports 24 to 32-pin EPROMs, several PLDs and all Philips PSD devices, members of the 8051 and 80C51 family of Microcontrollers
- Programs Lock Bits, Encryption Tables and Security Bits



PC-DOS, PC/XT/AT are trademarks of IBM.  
IBM is a registered trademark of IBM Corporation.  
Intel is a registered trademark of Intel Corporation.



# SPECIFICATIONS

## Supported Devices

**EPROMs:** 2716, 2732, 2764, 27128, 27256, 27512, 27010, 27040, both NMOS and CMOS versions for all the available programming voltages.

**Microcontrollers:** 8751H, 8751BH, 87C51, 87C51FA/B/C, 87C52, 87C451, 87C528, 87C550, 87C552, 87C562, 87C575, 87C592, 87C652, 87C654, 87C751, 87C752, 87C054-MT and others.

**PLDs:** AT22V10, ATV750, ATV2500, ATV5000.

**PSD Devices:** PSD3XX

## File Formats

MP-51 loads different file formats:

- Intel Hex files and Motorola S-records
- Binary files
- Object files
- JEDEC files

It saves portions of memory in Intel Hex, Binary and JEDEC formats.

## Command Set

The available functions include: TYPE, BLANK CHECK, SECURITY, PROGRAM, LOAD, SAVE, READ, VIEW, COMPARE, CHECKSUM, FILL, MOVE, MODIFY, DIRECTORY, CHANGE DIR, TEXT FILE, DUMP, QUIT.

## Host Characteristics

IBM PC/XT/AT or compatible system with 512 KBytes of RAM, one floppy disk drive, one RS-232 interface card for the PC, PC-DOS 2.0 or later.

## Input Power

85VAC to 263VAC, 50Hz to 60Hz. That makes this unit suitable for any country outlet.

## Mechanical Dimensions

MP-51 is 155mm long, 60mm high, and 250mm wide.

## Adapters and Supported Devices

The following list partially shows which adapter should be used for the different supported devices:

PPA-EPROM	28-Pin DIP	2716 to 27512 NMOS and CMOS
PPA-51X	40-Pin DIP	8751H, 8751BH, 87C51, 87C52, 87C528, 87C550, 87C652, 87C654
PPA-51XASD	44-Pin PLCC	8751H, 8751BH, 87C51, 87C51, 87C528, 87C652, 87C654
PPA-751SD	24-Pin Skinny DIP	87C751
PPA-751ASD	28-Pin PLCC	87C751
PPA-752SD	28-Pin DIP	87C752
PPA-752ASD	28-Pin PLCC	87C752
PPA-451SD	64-Pin DIP	87C451
PPA-451ASD	68-Pin PLCC	87C451
PPA-552ASD	68-Pin PLCC	87C552, 87C562
PPA-550ASD	44-Pin PLCC	87C550-PLCC
PPA-592ASD	68-Pin PLCC	87C592
PPA-054SD	42-Pin Shrink DIP	87C054-MTV

## Items Supplied as Standard

MP-51 Programmer. User software. User's Manual. RS-232 cable and adapters. Power cord is not included.

## Warranty

Six months limited warranty; parts and labor.

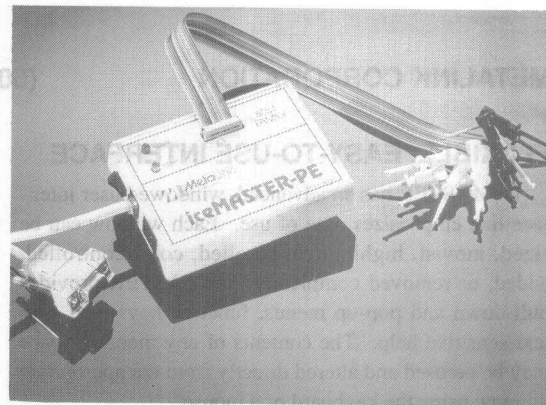
## Ordering Information

**PMP-51 SD (includes PPA-51X adapter)**

Available through Philips distributors.

# iceMASTER-PE™ 8051 Family In-Circuit Emulator

NOW  
8051, 8052,  
8xC51FX  
SUPPORT



## PRODUCT DESCRIPTION

The unique **iceMASTER-PE** packs an advanced feature set into a tiny, palm-sized package that any engineer can afford. Designed for demanding projects, **iceMASTER-PE** supports frequencies up to 42 MHz with a full complement of emulation memory, external data memory, and a transparent trace buffer 16K frames deep with advanced searching capabilities. The entire emulator plugs directly into the target application or operates in a stand-alone mode.

The **iceMASTER-PE** is the world's most portable emulator because the emulator and probe electronics are both integrated into a pocket-able package about the size of a PC mouse. To achieve this dramatic breakthrough in the size and cost of high-performance emulation, MetaLink invented a new emulation system architecture, Advanced Emulator Technology (AET, patent pending).

The **iceMASTER-PE**'s windowed user interface delivers the highest development productivity and is easy to learn and easy to use, including a context-sensitive hypertext and hyperlinked help system. This powerful, productive interface gives the user total control and flexibility in the configuration of the size, position, content, and color of each window.

The **iceMASTER-PE** includes a full symbolic and source-level debugger for Assemblers and Compilers. The emulator supports all 9 of the most popular 8051 Assemblers and Compilers.

The **iceMASTER-PE** sets the new standard for excellence, portability and value in 8051 family emulation.

## KEY CHARACTERISTICS

- ☐ Supports 8051 family devices up to 42MHz
- ☐ Support for 8031, 8051, 8x32, 8x52, 8xC54, 8xC58, 8xC51FA, 8xC51FB, 8xC51FC, 8xC751, 8xC752, 8xC750, NMOS, CMOS, ROM, EPROM/OTP Devices.
- ☐ Full-Featured, Real-time & Transparent Emulator
- ☐ Emulator and Probe electronics integrated into a single package only 3" x 4" x 1"
- ☐ Plugs directly into target applications or operates in a stand-alone mode
- ☐ Patent-pending AET system architecture

## HARDWARE CAPABILITIES

- ☐ Up to 64K Program & 64K External Data Memory
- ☐ 16K frame trace buffer
- ☐ View trace while executing
- ☐ Up to 128K hardware breakpoints
- ☐ Up to 64K Trace ON & 64K Trace OFF triggers
- ☐ Integrated diagnostic self-test capability

## SYSTEM FEATURES

- ☐ PC-hosted via RS-232 serial link
- ☐ Efficient, powerful, easy to learn
- ☐ User control of window size, content & color
- ☐ Supports third party Assemblers & Compilers
- ☐ Full Symbolic & Source-Level debug
- ☐ Complete system includes Emulator, 8051 Macro Cross Assembler, RS-232 cable & power supply

MetaLink Corporation  
325 E. Elliot Road  
Chandler, AZ 85225  
Phone: (602) 926-0797  
Phone: (800) 638-2423  
Fax: (602) 926-1198

MetaLink Europe GmbH  
8011 Kirchseeon-Eghharting  
Germany  
Phone: 08091 2046  
Fax: 08091 2386

© 1993 MetaLink Corporation

1042-001/692/TP

**MetaLink®**  
Corporation

The screenshot shows the PC BIOS setup screen. The 'Load' menu is open, displaying the following options:

- Load Code Memory from a File:
- File: F:\demo.
- Directory: ? : terse ! : verbose
- Esc:Cancel

The background shows the BIOS main menu with the following options:

- File Run Display/Alter Misc Source/Symbols Break/Trace Help
- Module-F
- 00EF C3
- 00FF 8520
- 00FF 6400
- 00FF 9400
- 00FF 5004
- F.WAS:R:412
- 00FF 8523
- 00FF 80F3
- Keyboard Input: Control & Number/Symbol Syntax
- Keyboard Editing Control Keys
- General
- Help for Help
- Ctrl-B:Resize Ctrl-V:Move
- Esc:Exit
- CC 00
- .....
- SP:28
- ret add=00CCh
- 0052 (D0M0)
- C:0 F0B:0 04:0 U: F:8
- J0C0N ..00 !:---C3
- FSW...00 K0R2N:00
- int DPTA:0000 PC:00FC
- Exit:Addr:00FC
- .....
- SP:28

The 'Load' menu is currently selected, and the 'File' field shows 'F:\demo.'.

```

Bank 0
R0:R1 R2:R3 R4:R5 R6:R7 R8:R9 R10:R11 R12:R13 R14:R15 R16:R17 R18:R19 R20:R21 R22:R23 R24:R25 R26:R27 R28:R29 R30:R31
AC-----0 B-----0 DPH-----0 DPL-----0 IE-----0 IOCCN-----0 IP-----0
+ 00-----FF 01-----FF P2-----FF P3-----FF P4-----0 P5-----0 NCPN23H-0

Registers
0052 (D0H0)
0053 (D0H1)
0054 (D0H2)
0055 (D0H3)
0056 (D0H4)
0057 (D0H5)
0058 (D0H6)
0059 (D0H7)
0060 (D0H8)
0061 (D0H9)
0062 (D0HA)
0063 (D0HB)
0064 (D0HC)
0065 (D0HD)
0066 (D0HE)
0067 (D0HF)
0068 (D0HG)
0069 (D0HH)
0070 (D0HI)
0071 (D0HJ)
0072 (D0HK)
0073 (D0HL)
0074 (D0HM)
0075 (D0HN)
0076 (D0HO)
0077 (D0HP)
0078 (D0HQ)
0079 (D0HR)
0080 (D0HS)
0081 (D0HT)
0082 (D0HU)
0083 (D0HV)
0084 (D0HW)
0085 (D0HX)
0086 (D0HY)
0087 (D0HZ)
0088 (D0I0)
0089 (D0I1)
0090 (D0I2)
0091 (D0I3)
0092 (D0I4)
0093 (D0I5)
0094 (D0I6)
0095 (D0I7)
0096 (D0I8)
0097 (D0I9)
0098 (D0IA)
0099 (D0IB)
0100 (D0IC)
0101 (D0ID)
0102 (D0IE)
0103 (D0IF)
0104 (D0IG)
0105 (D0IH)
0106 (D0II)
0107 (D0IJ)
0108 (D0IK)
0109 (D0IL)
0110 (D0IM)
0111 (D0IN)
0112 (D0IO)
0113 (D0IP)
0114 (D0IQ)
0115 (D0IR)
0116 (D0IS)
0117 (D0IT)
0118 (D0IU)
0119 (D0IV)
0120 (D0IW)
0121 (D0IX)
0122 (D0IY)
0123 (D0IZ)
0124 (D0J0)
0125 (D0J1)
0126 (D0J2)
0127 (D0J3)
0128 (D0J4)
0129 (D0J5)
0130 (D0J6)
0131 (D0J7)
0132 (D0J8)
0133 (D0J9)
0134 (D0JA)
0135 (D0JB)
0136 (D0JC)
0137 (D0JD)
0138 (D0JE)
0139 (D0JF)
0140 (D0JG)
0141 (D0JH)
0142 (D0JI)
0143 (D0JJ)
0144 (D0JK)
0145 (D0JL)
0146 (D0JM)
0147 (D0JN)
0148 (D0JO)
0149 (D0JP)
0150 (D0JQ)
0151 (D0JR)
0152 (D0JS)
0153 (D0JT)
0154 (D0JU)
0155 (D0JV)
0156 (D0JW)
0157 (D0JX)
0158 (D0JY)
0159 (D0JZ)
0160 (D0K0)
0161 (D0K1)
0162 (D0K2)
0163 (D0K3)
0164 (D0K4)
0165 (D0K5)
0166 (D0K6)
0167 (D0K7)
0168 (D0K8)
0169 (D0K9)
0170 (D0KA)
0171 (D0KB)
0172 (D0KC)
0173 (D0KD)
0174 (D0KE)
0175 (D0KF)
0176 (D0KG)
0177 (D0KH)
0178 (D0KI)
0179 (D0KJ)
0180 (D0KK)
0181 (D0KL)
0182 (D0KM)
0183 (D0KN)
0184 (D0KO)
0185 (D0KP)
0186 (D0KQ)
0187 (D0KR)
0188 (D0KS)
0189 (D0KT)
0190 (D0KU)
0191 (D0KV)
0192 (D0KW)
0193 (D0KX)
0194 (D0KY)
0195 (D0KZ)
0196 (D0L0)
0197 (D0L1)
0198 (D0L2)
0199 (D0L3)
0200 (D0L4)
0201 (D0L5)
0202 (D0L6)
0203 (D0L7)
0204 (D0L8)
0205 (D0L9)
0206 (D0LA)
0207 (D0LB)
0208 (D0LC)
0209 (D0LD)
0210 (D0LE)
0211 (D0LF)
0212 (D0LG)
0213 (D0LH)
0214 (D0LI)
0215 (D0LJ)
0216 (D0LK)
0217 (D0LL)
0218 (D0LM)
0219 (D0LN)
0220 (D0LO)
0221 (D0LP)
0222 (D0LQ)
0223 (D0LR)
0224 (D0LS)
0225 (D0LT)
0226 (D0LU)
0227 (D0LV)
0228 (D0LW)
0229 (D0LX)
0230 (D0LY)
0231 (D0LZ)
0232 (D0M0)
0233 (D0M1)
0234 (D0M2)
0235 (D0M3)
0236 (D0M4)
0237 (D0M5)
0238 (D0M6)
0239 (D0M7)
0240 (D0M8)
0241 (D0M9)
0242 (D0MA)
0243 (D0MB)
0244 (D0MC)
0245 (D0MD)
0246 (D0ME)
0247 (D0MF)
0248 (D0MG)
0249 (D0MH)
0250 (D0MI)
0251 (D0MJ)
0252 (D0MK)
0253 (D0ML)
0254 (D0MM)
0255 (D0MN)
0256 (D0MO)
0257 (D0MP)
0258 (D0MQ)
0259 (D0MR)
0260 (D0MS)
0261 (D0MT)
0262 (D0MU)
0263 (D0MV)
0264 (D0MW)
0265 (D0MX)
0266 (D0MY)
0267 (D0MZ)
0268 (D0N0)
0269 (D0N1)
0270 (D0N2)
0271 (D0N3)
0272 (D0N4)
0273 (D0N5)
0274 (D0N6)
0275 (D0N7)
0276 (D0N8)
0277 (D0N9)
0278 (D0NA)
0279 (D0NB)
0280 (D0NC)
0281 (D0ND)
0282 (D0NE)
0283 (D0NF)
0284 (D0NG)
0285 (D0NH)
0286 (D0NI)
0287 (D0NJ)
0288 (D0NK)
0289 (D0NL)
0290 (D0NM)
0291 (D0NN)
0292 (D0NO)
0293 (D0NP)
0294 (D0NQ)
0295 (D0NR)
0296 (D0NS)
0297 (D0NT)
0298 (D0NU)
0299 (D0NV)
0300 (D0NW)
0301 (D0NX)
0302 (D0NY)
0303 (D0NZ)
0304 (D0O0)
0305 (D0O1)
0306 (D0O2)
0307 (D0O3)
0308 (D0O4)
0309 (D0O5)
0310 (D0O6)
0311 (D0O7)
0312 (D0O8)
0313 (D0O9)
0314 (D0OA)
0315 (D0OB)
0316 (D0OC)
0317 (D0OD)
0318 (D0OE)
0319 (D0OF)
0320 (D0OG)
0321 (D0OH)
0322 (D0OI)
0323 (D0OJ)
0324 (D0OK)
0325 (D0OL)
0326 (D0OM)
0327 (D0ON)
0328 (D0OO)
0329 (D0OP)
0330 (D0OQ)
0331 (D0OR)
0332 (D0OS)
0333 (D0OT)
0334 (D0OU)
0335 (D0OV)
0336 (D0OW)
0337 (D0OX)
0338 (D0OY)
0339 (D0OZ)
0340 (D0P0)
0341 (D0P1)
0342 (D0P2)
0343 (D0P3)
0344 (D0P4)
0345 (D0P5)
0346 (D0P6)
0347 (D0P7)
0348 (D0P8)
0349 (D0P9)
0350 (D0PA)
0351 (D0PB)
0352 (D0PC)
0353 (D0PD)
0354 (D0PE)
0355 (D0PF)
0356 (D0PG)
0357 (D0PH)
0358 (D0PI)
0359 (D0PJ)
0360 (D0PK)
0361 (D0PL)
0362 (D0PM)
0363 (D0PN)
0364 (D0PO)
0365 (D0PP)
0366 (D0PQ)
0367 (D0PR)
0368 (D0PS)
0369 (D0PT)
0370 (D0PU)
0371 (D0PV)
0372 (D0PW)
0373 (D0PX)
0374 (D0PY)
0375 (D0PZ)
0376 (D0Q0)
0377 (D0Q1)
0378 (D0Q2)
0379 (D0Q3)
0380 (D0Q4)
0381 (D0Q5)
0382 (D0Q6)
0383 (D0Q7)
0384 (D0Q8)
0385 (D0Q9)
0386 (D0QA)
0387 (D0QB)
0388 (D0QC)
0389 (D0QD)
0390 (D0QE)
0391 (D0QF)
0392 (D0QG)
0393 (D0QH)
0394 (D0QI)
0395 (D0QJ)
0396 (D0QK)
0397 (D0QL)
0398 (D0QM)
0399 (D0QN)
0400 (D0QO)
0401 (D0QP)
0402 (D0QQ)
0403 (D0QR)
0404 (D0QS)
0405 (D0QT)
0406 (D0QU)
0407 (D0QV)
0408 (D0QW)
0409 (D0QX)
0410 (D0QY)
0411 (D0QZ)
0412 (D0R0)
0413 (D0R1)
0414 (D0R2)
0415 (D0R3)
0416 (D0R4)
0417 (D0R5)
0418 (D0R6)
0419 (D0R7)
0420 (D0R8)
0421 (D0R9)
0422 (D0RA)
0423 (D0RB)
0424 (D0RC)
0425 (D0RD)
0426 (D0RE)
0427 (D0RF)
0428 (D0RG)
042
```

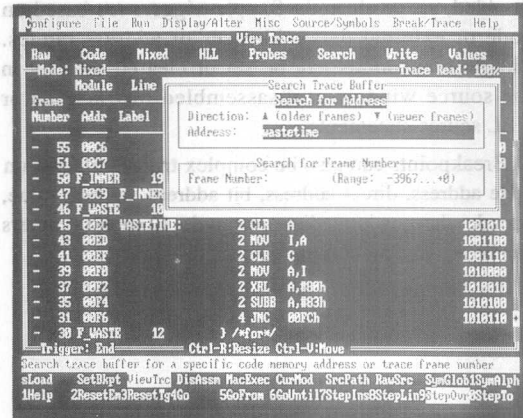
## THE COMPLETE DEBUGGER

The trace buffer captures data in real-time. Trace information consists of address and data bus values and user-selectable probe clips. You can view the trace buffer data through several display filters: raw hex, disassembled instructions, instructions mixed with HLL source statements, or HLL source only. You can display the probe clip bit values in binary, hex, or digital waveform formats.

You can trigger the trace to begin capturing data on all instructions leading up to a breakpoint, around (before and after) a breakpoint, or following a breakpoint. Capture filtering allows you to focus attention only on areas of interest, eliminating clutter.

To further speed your design process, an integrated search mechanism allows you to locate any label, HLL source line number, or address in the trace buffer in either the backward or forward direction. The days of manually scanning or post-processing a large buffer of trace data are gone!

If you WRITE your program in a HLL, you should be able to DEBUG it that way — iceMASTER lets you do just that!

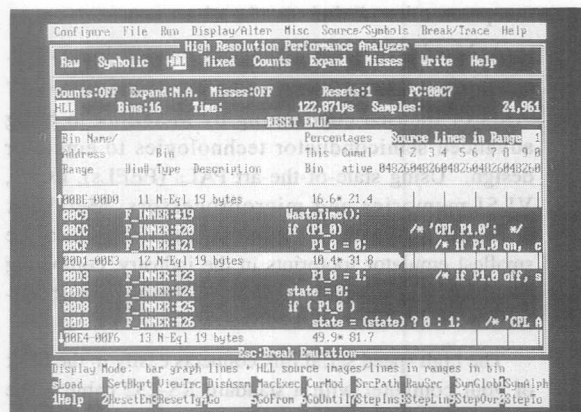


## IMPROVING THE QUALITY OF YOUR PRODUCT: FINDING THE HOT (OR NOT SO HOT) SPOTS

iceMASTER emulators (Model 400 & PE) provide a PERFORMANCE ANALYZER that allows you to monitor the time spent executing specific portions of your program to find "hot spots" or "dead code." You can define and analyze memory areas based on code address, module, line number or label ranges.

You can view results dynamically during emulation or later for a more detailed analysis. You can toggle the display from bar graph format to actual frequency counts, and you can filter the level of detail in the displayed results to include raw data, code labels, and/or HLL source statements.

iceMASTER Model 400 provides an additional Performance Analyzer that is the absolute best in the industry for tuning and testing your code. It is very flexible and far more accurate than most other emulators, with a resolution of less than six microseconds. You can define and analyze up to 15 memory areas and each of these 15 memory areas can consist of non-contiguous, logically related subranges, (e.g. groupings of related functions which are not necessarily contiguous in memory).



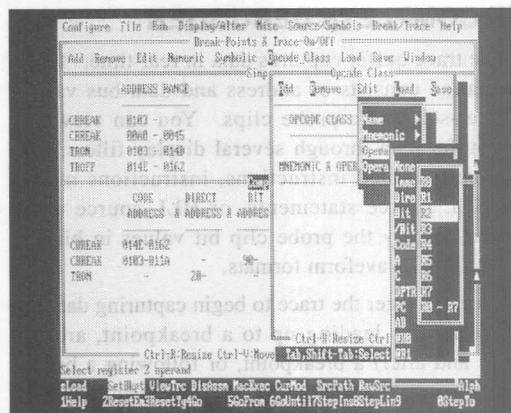


## FINDING THAT BUG

You have access to as many as 128K breakpoints and 64K tracetriggers. These triggers can be enabled, disabled, set or cleared. Simple triggers are based on code or external data addresses or address ranges. Simple breakpoints can be set or cleared directly in the source window at disassembled instructions or HLL source statements.

Breakpoints can also be complex triggers, based on code address, direct address, bit address, opcode value, opcode class or immediate operand. Complex triggers can be ANDed and ORed together.

Finding that elusive bug is now much easier!



To further speed your design process, an integrated search mechanism allows you to locate any label, HLL source line number, or address in the trace buffer in either the backward or forward direction. The days of manually scanning or post-processing a large buffer of trace data are gone!

If you WRITE your program in a HLL, you should be able to DEBUG it that way - iceMASTER lets you do just that!

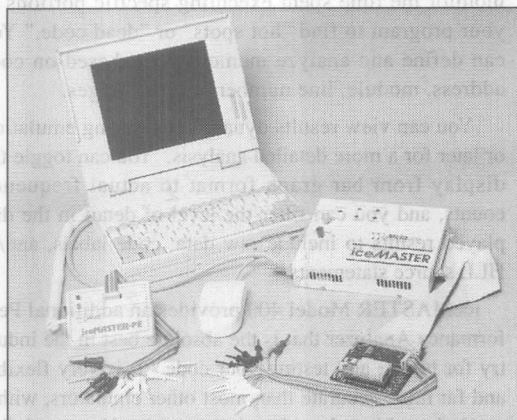
## SMALL AND POWERFUL

Some companies design emulators that use one or two full size PC expansion board slots. These companies take it for granted that you have the "right" kind of PC, that you don't need the expansion slots for something else, and that you don't mind taking apart your computer to install their emulator!

At MetaLink, we don't take our customers for granted.

The iceMASTER family is the culmination of over 7 years of focused engineering by MetaLink to bring advanced semiconductor technologies to emulator design. Using state-of-the-art PALs (PEELs), LCAs, VLSI memories and microprocessors, MetaLink designed world-class emulation capability into the smallest emulator footprints in the industry, combining powerful and complex emulation features with all-around ease-of-use.

The high speed serial link connects easily to the back of your computer with a standard RS-232 cable. The emulators are smaller than the size of a VCR tape, fit neatly in crowded work spaces and are easy to move around on the bench or to other computers. **PORTABLE PRODUCTIVITY!**





# iceMASTER-PE Development System

MetaLink specializes in enhancing the emulation technology required by EMBEDDED SYSTEMS DESIGNERS. MetaLink has consistently led the industry in emulation technology: The first PC-based 8051 emulator; the first 8052 emulator; the first to offer support for all the unique features of the 8051 family components, such as watchdog-timer, idle modes, power down mode, DMA and A/D.

MetaLink is a full-service emulation company. We support our customers over the long term with services such as repair, discounted upgrades, rental units, 10-day trial purchase periods and free technical support for applications problems. A network of world-wide sales and service representatives is augmented by a well-trained telemarketing staff at headquarters.

## ❑ HOST SPECIFICATIONS

Host IBM PC, XT, AT, 386, 486, PS/2, Laptop, Notebook or compatible system, 640K bytes of RAM, Hard Disk Drive, Monochrome or Color Display, with a Standard RS-232 Serial Port. Operating System DOS 2.0 or greater.

## ❑ EMULATOR SOFTWARE SPECIFICATIONS

File Formats Supported with Symbolic or Source-Level Debugging:

2500AD, Archimedes, Avocet, BSO/Tasking, Franklin, IAR Systems, Intel OMF, Keil, MCC, Microtec Research, Motorola 'S' Record, Systonix and Intel HEX.

Source/Symbol Support:

Assembler, BASIC, C and PL/M Languages

Symbolic or Source-Level Debug:

Setting of Breakpoints  
Setting of Trace ON/OFF  
Viewing of Trace Buffer  
Viewing of Source Window Display  
Viewing of Performance Analyzer  
Assembly/Disassembly of Code

HLL Structure/Content Display of:

Modules Line Numbers  
Scopes Program Variables

## ❑ EMULATOR HARDWARE SPECIFICATIONS

iceMASTER-PE MODELS:

8031: Supports 8031 and 80C31  
Operates up to 33MHz  
40-Lead DIP or optional 44-Lead PLCC

8032: Supports 8031, 80C31, 8032, 80C32 and C501  
Operates up to 42MHz  
40-Lead DIP or optional 44-Lead PLCC

83752: Supports 83C751, 87C751, 83C752, 87C752 and 87C750  
Operates up to 16MHz  
24- & 28-Lead DIP or optional 28-Lead PLCC

8351FX: Supports 8031, 80C31, 8032, 80C32, 8051, 80C51, 8751, 87C51, 8052, 80C52, 8752, 87C52, 80C51FA, 83C51FA, 87C51FA, 83C51FB, 87C51FB, 83C51FC, 87C51FC, 80C54, 87C54, 80C58 and 87C58  
Operates up to 16MHz  
40-Lead DIP or optional 44-PLCC

## ❑ OPERATING CHARACTERISTICS

Clock frequency user-selectable between external target crystal/clock or internal clock source  
Real-time, Electrically and Operationally Transparent

## ❑ USER INTERFACE

Keyboard or Mouse Control

Pull-down & Pop-up menus with fill-in boxes

Available Main Screen Windows:

Registers and PSW bits  
Bit Memory  
Stack data displayed in HEX &/or ASCII  
Up to 5 Internal Data Memory displayed in HEX &/or ASCII  
Up to 5 External Data Memory displayed in HEX &/or ASCII  
Up to 5 Code Memory displayed in HEX &/or ASCII  
Source Program displayed in Code, HLL Source or Mixed  
Watch window for variable data  
System Status data

Main Screen Window Display Controls:

Movable Selectable (On/Off)  
Sizable Color selection  
Scrollable Highlighting of key data

Function/Hot Key Access:

User-assignable for commands  
User-displayable for quick reference

## ❑ MEMORY OPERATIONS

Emulation Memory:

iceMaster-PE Models: 8031 8032 8351FX 83752

Program Memory: 64K 64K 64K 2K

External Data Memory: 64K 64K 64K N/A

Mapping Resolution:

Program: Down to 1-byte  
External Data: Down to 1-byte

Program Memory:

Single Line Assembler (full instruction set support)

Disassemble in Code or Source/Code mode

Disassembly may be written to a disk file

Data Memory:

Internal or External Memory

Fill a block of memory with data

Copy a block of data to another area

Change a single address data content

Compare any two blocks of addressed data

Displayed data may be written to a file

Registers/SFRs/Bit Memory:

Examine or Modify

Program Variables:

Examine or Modify

## ❑ EMULATION CONTROLS

Reset from Emulator and Go

Reset from Target and Go

Reset Processor

Go from current Program Counter

Go From a new Program Counter

Go Until a Program Counter/Label

Slow Motion (Repetitive Step commands)

Step by machine instruction

Step by Line Number

Step Over calls

Step To next function or procedure

## ❑ HARDWARE BREAKPOINTS

Up to 64K real-time Program Addresses

Up to 64K real-time External Data Addresses

## ❑ TRIGGER CONDITIONS

Set directly in Source window or Pull-down menu

PC address & range of addresses

Opcode Value

Opcode Class

SFRs/Registers

Direct byte address & range of addresses

Direct bit address & range of addresses

Immediate operand value

Read/Write to bit address

Register address modes

Read/Write to Register address

Logical AND/OR of any of the above

External Data address & range of addresses

Break Count Overflow

External (CLIP) Break Input

External (CLIP) Trigger Output

## ❑ TRACE

Real-time trace with view while executing code

16K-Frame Trace Buffer

Start, Center, End and Variable Trace Trigger settings

Up to 64K Trace ON/OFF triggers for trace filtering

Trace Contents consist of:

16-bits Address Bus

8-bits Data Bus

7-bits External Clips

Trace Display Modes:

Raw Hex

Symbolic

HLL Source

Mixed

External Clips display format:

Binary data

HEX data

Digital waveform

Trace Buffer Operations:

Write trace buffer to a disk file

Search trace buffer for labels & addresses

## ❑ PERFORMANCE ANALYZER

Program profiling capability

7 year duration

Display options:

Bar Graph

Frequency Count

Display Modes:

Raw

HLL Source Lines

Symbolic Mixed

Up to 999 Bin capacity

User-controlled Bin set-up:

By Address

By Symbol

By Module

By Line Number

Automatic

## ❑ HELP

On-Line

Context sensitive

Hypertext/Hyperlinked

## ❑ DIAGNOSTIC SELF-TEST

Determines Emulation Hardware Status

## ❑ MACRO

Repetitive routines

User-created and callable

## ❑ ELECTRICAL SPECIFICATIONS

Input Power (maximum):

0.75 A @ +5 VDC +/-5%

Power Source: Target system or power supply

## ❑ MECHANICAL SPECIFICATIONS

Emulator Dimensions:

4.4" x 3.25" x 0.9"

11.18cm x 8.25cm x 2.29cm

Emulator weight:

0.5lbs 0.3kg

## ❑ ANNUNCIATORS

Power and Active LEDs

Power ON/OFF switch

## ❑ WARRANTY

One (1) year limited warranty, parts and labor

Call Today

1(800) 638-2423

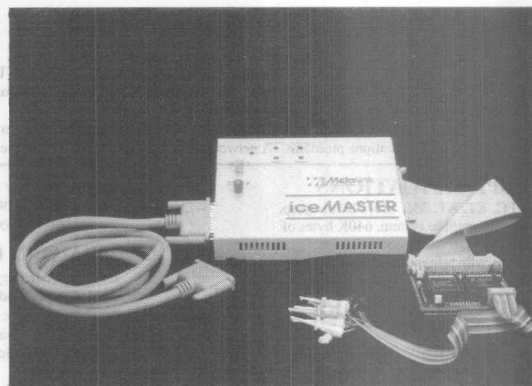
1(800) METAICE

or contact your local distributor

Rental plans are available.

Product names are used to purposes of identification only and may be trademarks or registered trademarks of their respective companies.

# iceMASTER™ 8051 In-Circuit Emulators



## PRODUCT DESCRIPTION

The **iceMASTER-8051 Model 200/400** emulator represents a culmination of 7 years of focused engineering to create the world's most affordable and portable full-featured emulators.

The **iceMASTER-8051** emulators offer real-time and transparent emulation at up to 24MHz for a broad number of derivative devices through the use of interchangeable probe cards. Powerful breakpoint systems allow an engineer to stop a program at any time and examine all states and conditions. Trace memory provides a complete history of each event that has occurred, including source-level information, address, data, status, searching and external logic events. The best performance analyzer capability in the industry allows a thorough evaluation of the program to decide what areas are taking the most time and simplify those areas requiring improved performance.

The **iceMASTER-8051** emulators support symbolic and source-level debugging for the most popular 8051 Cross Assemblers, 'C' Compilers, and PL/M Compilers. These capabilities allow the designer to debug their system the way it was designed, at the symbolic or source-level. This methodology increases productivity while decreasing cost and time-to-market.

The **iceMASTER-8051** emulators provide pull-down and pop-up menus, mouse support, function/hot keys, and context-sensitive hyperlinked help. The advanced windowed user interface emphasizes ease of use. Each window can be sized, moved, scrolled, highlighted, color-controlled, added or removed completely. The contents of any memory space may be perused and altered directly from the appropriate window, with multiple memory spaces displayable simultaneously.

## KEY CHARACTERISTICS

- ❑ Full-Featured, Real-time & Transparent Emulator
- ❑ Supports 8051 family devices up to 24MHz
- ❑ Interchangeable Probe Cards
- ❑ Hosted on any PC or compatible, including Laptops, Notebooks, or Micro Channel
- ❑ 115K baud serial link using a standard comm port
- ❑ Unlimited user support

## HARDWARE CAPABILITIES

- ❑ 64K Program & 64K External Data Memory
- ❑ 4K frame trace buffer
- ❑ Advanced trace search ability
- ❑ 128K hardware breakpoints
- ❑ 64K Trace ON/OFF triggers
- ❑ Broad 8051 derivative device support (more than 65 devices)
- ❑ Dual Performance Analyzers

## SYSTEM FEATURES

- ❑ Efficient, powerful, easy to learn
- ❑ User control of window size, content & color
- ❑ Supports third party Assemblers & Compilers
- ❑ Full Symbolic & Source-Level debug
- ❑ Complete system - includes Emulator, 8051 Macro Cross Assembler, RS-232 cable & power supply

MetaLink Corporation  
325 E. Elliot Road  
Chandler, AZ 85225

Phone: (602) 926-0797  
Phone: (800) 638-2423  
Fax: (602) 926-1198

MetaLink Europe GmbH  
8011 Kirchseeon-Eglharting  
Germany

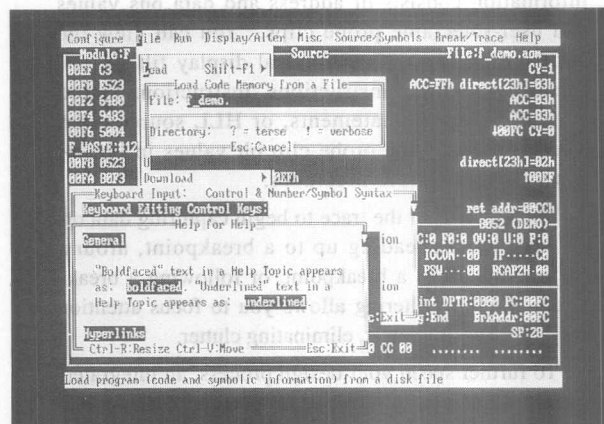
Phone: 08091 2046  
Fax: 08091 2386

## FLEXIBLE, EASY-TO-USE INTERFACE

iceMASTER has an advanced, windowed user interface that emphasizes ease of use. Each window can be sized, moved, highlighted, scrolled, color-controlled, added, or removed completely. iceMASTER provides pull-down and pop-up menus, function keys, and context-sensitive help. The contents of any memory space may be perused and altered directly from the appropriate window using the keyboard or a mouse.

You have immediate access to the hypertext/hyperlinked, context-sensitive, on-line help system which clearly explains what your options are (at any detail level you choose), keeping you productive. There is even a HELP-FOR-HELP feature. Whether you are beginning your first design project, or are a veteran designer searching for the fastest possible debugging method, you will appreciate the EASE-OF-USE features designed into iceMASTER.

Novices can navigate smoothly through a debugging session by accessing the commands and menus as standard pull-downs. Experienced designers can instantaneously pop-up their menu of choice by using redefinable hot keys.

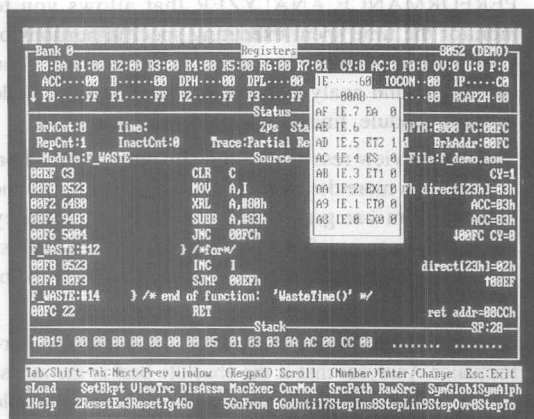


## DECREASED DEVELOPMENT TIME

Your iceMASTER source window accelerates the debugging process with Dynamically Annotated Code. When single-stepping, instruction execution information is displayed and retained next to each instruction. You can clearly see the data behind your program's flow, including contents of all accessed (read or write) memory locations and registers, as well as flow-of-control direction change markers. A moving color bar indicates the current position in the program as it executes.

At your option, the iceMASTER source window allows operations on three different views of code memory: disassembled instructions, instructions mixed with High Level Language (HLL) source statements, or HLL source only.

HLL source statements and symbolic disassembly information are also displayed when you disassemble the program or view the trace buffer.



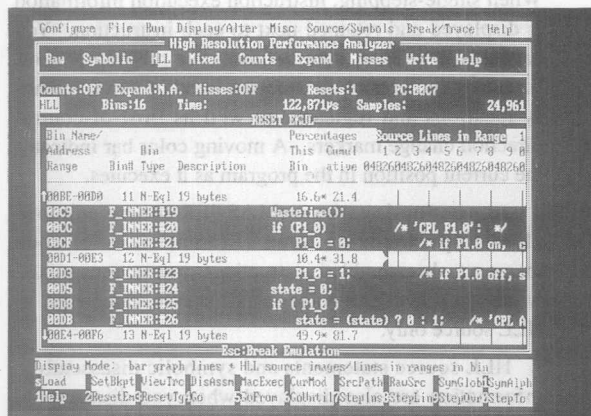
## THE COMPLETE DEBUGGER

The trace buffer captures data in real-time. Trace information consists of address and data bus values and user-selectable probe clips. You can view the trace buffer data through several display filters: raw hex, disassembled instructions, instructions mixed with HLL source statements, or HLL source only. You can display the probe clip bit values in binary, hex, or digital waveform formats.

You can trigger the trace to begin capturing data on all instructions leading up to a breakpoint, around (before and after) a breakpoint, or following a breakpoint. Capture filtering allows you to focus attention only on areas of interest, eliminating clutter.

To further speed your design process, an integrated search mechanism allows you to locate any label, HLL source line number, or address in the trace buffer in either the backward or forward direction. The days of manually scanning or post-processing a large buffer of trace data are gone!

If you WRITE your program in a HLL, you should be able to DEBUG it that way – iceMASTER lets you do just that!



## IMPROVING THE QUALITY OF YOUR PRODUCT: FINDING THE HOT (OR NOT SO HOT) SPOTS

iceMASTER emulators (Model 400 & PE) provide a PERFORMANCE ANALYZER that allows you to monitor the time spent executing specific portions of your program to find "hot spots" or "dead code." You can define and analyze memory areas based on code address, module, line number or label ranges.

You can view results dynamically during emulation or later for a more detailed analysis. You can toggle the display from bar graph format to actual frequency counts, and you can filter the level of detail in the displayed results to include raw data, code labels, and/or HLL source statements.

iceMASTER Model 400 provides an additional Performance Analyzer that is the absolute best in the industry for tuning and testing your code. It is very flexible and far more accurate than most other emulators, with a resolution of less than six microseconds. You can define and analyze up to 15 memory areas and each of these 15 memory areas can consist of non-contiguous, logically related subranges, (e.g. groupings of related functions which are not necessarily contiguous in memory).



**METALINK CORPORATION**

**(602) 926-0797**

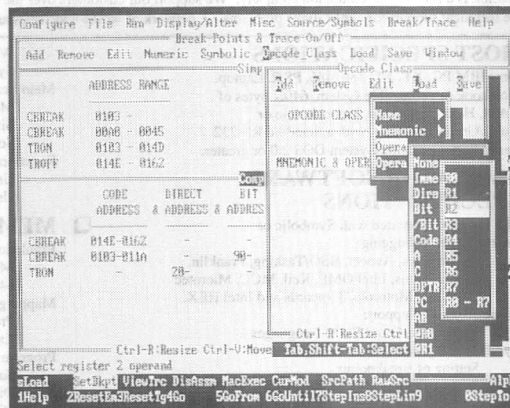
**FAX (602) 926-1198**

## FINDING THAT BUG

You have access to as many as 128K breakpoints and 64K tracetriggers. These triggers can be enabled, disabled, set or cleared. Simple triggers are based on code or external data addresses or address ranges. Simple breakpoints can be set or cleared directly in the source window at disassembled instructions or HLL source statements.

Breakpoints can also be complex triggers, based on code address, direct address, bit address, opcode value, opcode class or immediate operand. Complex triggers can be ANDed and ORed together.

Finding that elusive bug is now much easier!



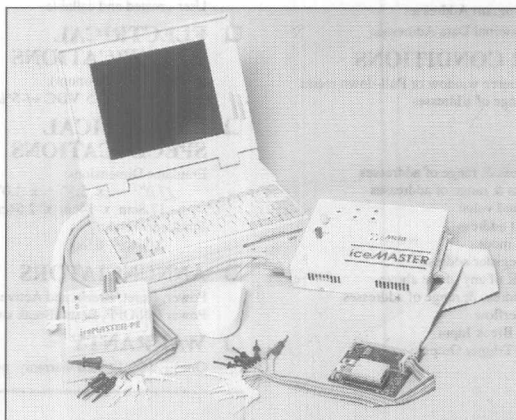
## SMALL AND POWERFUL

Some companies design emulators that use one or two full size PC expansion board slots. These companies take it for granted that you have the "right" kind of PC, that you don't need the expansion slots for something else, and that you don't mind taking apart your computer to install their emulator!

At MetaLink, we don't take our customers for granted.

The iceMASTER family is the culmination of over 7 years of focused engineering by MetaLink to bring advanced semiconductor technologies to emulator design. Using state-of-the-art PALs (PEELs), LCAs, VLSI memories and microprocessors, MetaLink designed world-class emulation capability into the smallest emulator footprints in the industry, combining powerful and complex emulation features with all-around ease-of-use.

The high speed serial link connects easily to the back of your computer with a standard RS-232 cable. The emulators are smaller than the size of a VCR tape, fit neatly in crowded work spaces and are easy to move around on the bench or to other computers. **PORTABLE PRODUCTIVITY!**





# iceMASTER-8051 Model 200/400 Development Systems

MetaLink specializes in enhancing the emulation technology required by EMBEDDED SYSTEMS DESIGNERS. MetaLink has consistently led the industry in emulation technology: The first PC-based 8051 emulator; the first 8052 emulator; the first to offer support for all the unique features of the 8051 family components, such as watchdog-timer, idle modes, power down mode, DMA and A/D.

MetaLink is a full-service emulation company. We support our customers over the long term with services such as repair, discounted upgrades, rental units, 10-day trial purchase periods, and free technical support for applications problems. A network of world-wide sales and service representatives is augmented by a well-trained telemarketing staff at headquarters.

## ❑ HOST SPECIFICATIONS

Host IBM PC, XT, AT, 386, 486, PS/2, Laptop, Notebook or compatible system, 640K bytes of RAM, Hard Disk Drive, Monochrome or Color/Graphic Display, with a Standard RS-232 Serial Port. Operating System DOS 2.0 or greater.

## ❑ EMULATOR SOFTWARE SPECIFICATIONS

File Formats Supported with Symbolic or

Source-Level Debugging:

Archimedes, Avocet, BSO/Tasking, Franklin, IAR Systems, Intel OMF, Keil, MCC, Microtec Research, Motorola 'S' records and Intel HEX.

Source/Symbol Support:

Assembler, C and PL/M Languages

Symbolic or Source-Level Debug:

Setting of Breakpoints

Setting of Trace ON/OFF

Viewing the Trace Buffer

Viewing of Source Window Display

Viewing of Performance Analyzer

Assembly/Disassembly of Code

HLL Structure/Content Display of:

Modules

Line Numbers

Scopes

Program Variables

## ❑ EMULATOR HARDWARE SPECIFICATIONS

iceMASTER-8051 Models:

200 Basic Emulator

400 Enhanced Emulator with:

4K Trace Buffer

2 Performance Analyzers

Full Watchdog Timer Support

Operating Modes:

Single-Chip ROM

Romless

Interchangeable Probe Card

Used to support the functional derivatives of each micro controller family as well as the full range of NMOS, CMOS, EPROM, and OTP technology variations. Contact MetalLink for the latest information on any device supported.

Device Unique Support

A-to-D Converter

FIFO

Multiple-Divide Unit

Multiple DPTRs

## ❑ OPERATING CHARACTERISTICS

Clock frequency user-selectable between external target crystal/clock or internal crystal source

Electrically and Operationally Transparent

Real-time: 0.5-24MHz

## ❑ USER INTERFACE

Keyboard or Mouse Control

Pull-down & Pop-up menus with fill-in boxes

Available Main Screen Windows:

Registers and PSW bits

Bit Memory

Stack data displayed in HEX &/or ASCII

Up to 5 Internal Data Memory displayed in HEX &/or ASCII

Up to 5 External Data Memory displayed in HEX &/or ASCII

Up to 5 Code Memory displayed in HEX &/or ASCII

Source Program displayed in Code, HLL Source or Mixed

Watch window for variable data

System Status data displayed in HEX

Main Screen Window Display Controls:

Movable

Selectable (On/Off)

Sizable

Color selection

Scrollable

Highlighting of key data

Function/Hot Key Access:

User-assignable for commands

User-displayable for quick reference

## ❑ MEMORY OPERATIONS

Emulation Memory:

64K Program Memory

64K External Data Memory

Mapping Resolution:

Program: 16-bytes/block

External Data: 16-bytes/block

Program Memory:

Single Line Assembler (full instruction set support)

Disassemble in Code or Source/Code mode

Disassembly may be written to a disk file

Data Memory:

Internal or External Memory

Fill of a block of memory with data

Copy a block of data to another area

Change a single address data content

Compare any two blocks of addressed data

Displayed data may be written to a file

Registers/SFRs/Bit Memory:

Examine or Modify

Program Variables:

Examine or Modify

## ❑ EMULATION CONTROLS

Reset from Emulator and Go

Reset from Target and Go

Reset Processor

Go from current Program Counter

Go From a new Program Counter

Go Until a Program Counter/Label

Slow Motion (Repetitive Step commands)

Step by machine instruction

Step by Line Number

Step Over calls

Step To next function or procedure

## ❑ HARDWARE BREAKPOINTS

64K real-time Program Addresses

64K real-time External Data Addresses

## ❑ TRIGGER CONDITIONS

Set directly in Source window or Pull-down menu

PC address & range of addresses

Opcode Value

Opcode Class

SFRs/Registers

Direct byte address & range of addresses

Direct bit address & range of addresses

Immediate operand value

Read/Write to bit address

Register address modes

Read/Write to Register address

Logical AND/OR of any of the above

External Data address & range of addresses

Break Count Overflow

External (CLIP) Break Input

External (CLIP) Trigger Output

## ❑ TRACE (Model 400)

4K-Frame Trace Buffer

Start, Center, End & Variable Trace Trigger settings

64K Trace ON/OFF triggers for trace filtering

Trace Contents consist of:

16-bits Address Bus

8-bits Data Bus

7-bits External Clips

DMA Activity

Trace Display Modes:

Raw Hex

Symbolic

HLL Source

Mixed

External Clips display format:

Binary data

HEX data

Digital waveform

Trace Buffer Operations:

Write trace buffer to a disk file

Search trace buffer for labels & addresses

## ❑ PERFORMANCE ANALYZERS (Model 400)

Real-time Program profiling capability

5.4μ-sec sampling period

7 year duration

Display options:

Bar Graph

Frequency Count

Display Modes:

Raw

Symbolic

Mixed

HLL Source Lines

Up to 999 Bin capacity

User-controlled Bin set-up:

By Address By Symbol

By Module By Line Number

Automatic

## ❑ HELP

On-Line

Context sensitive

Hypertext/Hyperlinked

## ❑ MACRO

Repetitive routines

User-created and callable

## ❑ ELECTRICAL SPECIFICATIONS

Input Power (maximum):

1.5 A @ +5 VDC +/-5%

## ❑ MECHANICAL SPECIFICATIONS

Emulator Dimensions:

7.0" x 5.5" x 1.0"

17.8cm x 14cm x 2.54cm

Emulator weight:

2.0lbs 0.9kg

## ❑ ANNUNCIATORS

Power, Reset, Break, and Active LEDs

Power ON/OFF, Reset, Break switches

## ❑ WARRANTY

One (1) year limited warranty, parts and labor



Call Today

1(800) 638-2423

1(800) METAICE or contact your local distributor

Rental plans are available.

Product names are used to purposes of identification only and may be trademarks or registered trademarks of their respective companies.

**Ordering Information****Philips Semiconductors 8051 Family Microcontroller Support****iceMASTER-PE Emulators**

The iceMASTER-PE emulators for the 8051 family are self-contained units, each dedicated to a particular member of the 8051 family. The emulator and probe card electronics are combined into a single, small package. Every iceMASTER-PE emulator comes with 64K of Code memory, 64K of External Data memory, a 16K frame Trace Buffer (which can be viewed without breaking emulation), power supply, RS-232 cable and an 8051 Macro Cross Assembler. All iceMASTER-PE emulators support an optional Break-Input signal clip, a Trigger-Output signal clip and seven user-placeable Trace-Capture-Input signal clips.

The following iceMASTER-PE emulators are currently available:

**PE-8031-33**

This emulator supports device operation from DC to 33MHz and has a 40-lead DIP footprint. This emulator supports the 8031 and 80C31 devices, regardless of process variation.

**PW-8032-30**

This emulator supports device operation from DC to 30MHz and has a 40-lead DIP footprint. This emulator supports the 8031, 80C31, 8032 and 80C32 devices, regardless of process variation.

**PE-83752-16**

This emulator supports device operation from DC to 16MHz and has a 28-lead DIP footprint. This emulator supports the 83C750, 87C750, 83C751, 87C751, 83C752 and 87C752 devices, regardless of process variation. This emulator is supplied with a 28-lead DIP to 24-lead DIP converter (CONV11) for 8xC751 support.

**PE-8351FX**

This emulator supports device operation from DC to 16MHz and has a 40-lead DIP footprint. This emulator supports the 8031, 80C31, 8032, 80C32, 8051, 80C51, 8751, 8052, 80C52, 8752, 87C52, 8xC51FX, 80C54, 87C54, 80C58 and 87C58, regardless of process variation.

The following iceMASTER-PE emulators will be available in Q2'94.

**PE-87L51FB**

This emulator supports device operation from 3.5 to 16MHz and has a 40-lead DIP footprint. This emulator supports the 87L51FB device from 3.0 to 5.5 volts.

**PE-8XCL410**

This emulator supports device operation from 0.5 to 16MHz and has a 40-lead DIP footprint. This emulator supports the 83CL410 and 80CL410 devices from 1.8 to 6 volts.

**PE-80CL51**

This emulator supports device operation from 32kHz to 16MHz and has a 40-lead DIP footprint. This emulator supports the 80CL51 and 80CL31 devices from 1.8 to 6 volts.

**iceMASTER Model 200 & Model 400 Emulators**

All iceMASTER Model 200 & Model 400 emulators support full probe card interchangeability. To emulate several different devices, you need purchase only a single emulator base unit, the iceMASTER-8051, and then one probe card for each unique device to be emulated.

Both the iceMASTER-8051 Model 200 and the iceMASTER-8051 Model 400 emulators come with 64K of Code memory and 64K of External Data memory, power supply, RS-232 cable and an 8051 Macro Cross Assembler.

The IceMASTER Model 400 emulator has the following additional features/capabilities:

1. 4K-Frame Trace Buffer
2. Performance Analyzers
  - a. High Resolution
  - b. Hig Bin Count
3. Full Watchdog Timer (WDT) Support in all emulation modes

**Philips-Signetics 8051 Family Probe Cards**

All MetaLink probe cards include a Break-Input signal clip, a Trigger-Output signal clip and seven user-placeable Trace-Capture-Input signal clips. In addition, there is a user-selectable jumper for XTAL source. Most probe cards have other user-selectable jumpers to control/configure the unique capabilities of each particular device. Some probe cards also contain a user-selectable jumper for Vcc source. iceMASTER-8051 emulators fully supports:

All I/O ports, with no restrictions.

The CMOS Idle and Power Down modes of operation, in those devices having such capabilities.

The Watchdog Timer (WDT), in those devices having such a capability. (iceMASTER-8051 Model 400 only)

Correct device interconnection footprints.

Following is a list of all Philips-Signetics 8051 Family Probe Cards which MetaLink currently offers. Since MetaLink is constantly adding to its list of supported devices, please contact MetaLink for information on any device not listed below.

If support for a particular device is available at different maximum frequencies (MHz), the description of the Probe Card appears only once, with the speed variations noted in the Probe Card names above the description. Example: 8031-24 for a 24MHz 8031 Probe Card and 8031-20 for a 20MHz 8031 Probe Card.

**MHW-8031-20****MHW-8031-24**

The 8031 Probe Card supports device operation from 0.5 to 24MHz and has a 40-lead DIP package interface. This probe card supports 8031 and 80C31 devices, regardless of process variation.

**MHW-8032-20**

The 8032 Probe Card supports device operation from 0.5 to 20MHz and has a 40-lead DIP package interface. This probe card supports 8031, 80C31, 8032, 80C32 and 80C32 devices, regardless of process variation.

**MHW-80C51FA-16**

The 80C51FA Probe Card supports device operation from 0.5 to 16MHz and has a 40-lead DIP package interface. This probe card supports 8031, 80C31, 8032, 80C51FA and 80C32 devices, regardless of process variation.

**MHW-83C51FX-16**

The 83C51FC Probe Card supports device operation from 0.5 to 16MHz and has either a 40-lead DIP or a 44-lead PLCC package interface. This probe card supports 8031, 80C31, 8032, 80C32, 8051, 80C51, 8751, 87C51, 8x51FA, 8x51FB, 8052, 80C52, 8752 and 87C52 devices, regardless of process variation.

**MHW-8052-16**

The 8052 Probe Card supports device operation from 0.5 to 16MHz and has a 40-lead DIP package interface. This probe card supports 8031, 80C31, 8032, 80C32, 8051, 80C51, 8052, 80C52, 8751, 87C51, 8752, and 87C52 devices, regardless of process variation.

**MHW-83C053-12**

The 83C053 Probe Card supports device operation from 6 to 12MHz and has a 42-lead Shrink DIP package interface. This probe card supports 83C053, 83C054, 87C054, 83C055 and 87C055 devices, regardless of process variation.

**MHW-80CL410-12**

The 80CL410 Probe Card supports device operation from 0.5 to 12MHz and has a 40-lead DIP package interface. This probe card supports the 80CL410 device (4.5V-5.5V only).

**MHW-80C451-16**

The 80C451 Probe Card supports device operation from 1.2 to 16MHz and has a 68-lead PLCC package interface. This probe card supports the 80C451 device, regardless of process variation. Converter #5 is available to convert the probe card to a 64-lead DIP footprint.

**MHW-83C451-12**

The 83C451 Probe Card supports device operation from 1.2 to 12MHz and has a 68-lead PLCC package interface. This probe card supports the 80C451, 83C451 and 87C451 devices, regardless of process variation. Converter #5 is available to convert the probe card to a 64-lead DIP footprint.

**MHW-80C528-16**

The 80C528 Probe Card supports device operation from 1.2 to 16MHz and has a 40-lead DIP package interface. This probe card supports the 80C528 device.

**MHW-83C528-12****MHW-83C528-16**

The 83C528 Probe Card supports device operation from 1.2 to 16MHz and has either a 40-lead DIP or 44-lead PLCC package interface. This probe card supports 80C528, 83C528, 87C528, 83C524 and 87C524 devices, regardless of process variation.

#### **MHW-83C550-12**

The 83C550 Probe Card supports device operation from 1.2 to 12MHz and has a 44-lead PLCC package interface. This probe card supports the 83C550, 80C550 and 87C550 devices, regardless of process variation.

#### **MHW-80C552-16**

The 80C552 Probe Card supports device operation from 1.2 to 16MHz and has a 68-lead PLCC package interface. This probe card supports the 80C552 and 80C562 devices, regardless of process variation. Converter #7 is available to convert the probe card interface to a 40-lead DIP footprint for operation as an 8031, 8032, 80C31, 80C32 or 80C652.

#### **MHW-83C552-16**

The 83C552 Probe Card supports device operation from 1.2 to 16MHz and has a 68-lead PLCC package interface. This probe card supports the 80C552, 83C552, 87C552, 80C562, 83C562 and 87C562 devices, regardless of process variation. Converter #7 is available to convert the probe card interface to a 40-lead DIP footprint for operation as an 8031, 8032, 80C31, 80C32, 8x51, 8xC51, 8x52, 8xC52, 8xC652 or 8xC654.

#### **MHW-83C575-16**

The 83C575 Probe Card supports device operation from 0.5 to 16MHz and has either a 40-lead DIP or a 44-lead PLCC package interface. This Probe Card supports 80C575, 83C575 and 87C575 devices, regardless of process variation.

#### **MHW-83C592-16**

The 83C592 Probe Card supports device operation from 1.2 to 16MHz and has a 68-lead PLCC package interface. This probe card supports the 80C592, 83C592 and 87C592 devices, regardless of process variation.

#### **MHW-80C652-16**

The 80C652 Probe Card supports device operation from 1.2 to 16MHz and has a 40-lead DIP package interface. This probe card supports 8031, 80C31 and 80C652 devices from any IC manufacturer, regardless of process variation.

#### **MHW-83C652-16**

The 83C652 Probe Card supports device operation from 1.2 to 16MHz and has a 40-lead DIP package interface. This probe card supports 80C652, 83C652 and 87C652 devices from any IC manufacturer, regardless of process variation.

#### **MHW-83C654-16**

The 83C654 Probe Card supports device operation from 1.2 to 16MHz and has a 40-lead DIP package interface. This probe card supports 80C652, 83C654 and 87C654 devices from any IC manufacturer, regardless of process variation.

#### **MHW-83C751-16**

The 83C751 Probe Card supports device operation from 0.5 to 16MHz and has 24-lead Skinny DIP package interface. This probe card supports 83C751 and 87C751 devices, regardless of process variation.



**MHW-83C752-12**

The 83C752 Probe Card supports device operation from 0.5 to 12MHz and has a 28-lead DIP package interface. This probe card supports 83C752 and 87C752 devices, regardless of process variation.

**MHW-80C851-12**

The 80C851 Probe Card supports device operation from 1.2 to 12MHz and has a 40-lead DIP package interface. This probe card supports 8031, 80C31, and 80C851 devices, regardless of process variation.

**MHW-83C851-12**

The 83C851 Probe Card supports device operation from 1.2 to 12MHz and has a 40-lead DIP package interface. This probe card supports 80C851, and 83C851 devices, regardless of process variation.

**iceMASTER Converters**

iceMASTER converters are used with the iceMASTER-8051 Probe Card to allow the user to change the device footprint supported by the Probe Card to a new device footprint or to allow the iceMASTER-8051 Probe Card to support a new device with the existing Probe Card.

**MHW-CONV5**

A 68-lead PLCC to 64-lead DIP converter for 83C451 or 80C451 Probe Cards to 80C451, 83C451 and 87C451 devices.

**MHW-CONV7**

A 68-lead PLCC to 40-lead DIP converter for 83C552 or 80C552 Probe Cards to 8031, 8032\*, 80C31, 80C32\*, 8051, 8751, 80C51, 87C51, 8052\*, 8752\*, 80C52\*, 80C652, 83C652, 87C652, 87C654, 83C654 and 87C52\* devices. \*Not all modes of Timer 2 supported.

**MHW-CONV11**

A 28-lead DIP to 24-lead DIP converter for the PE-83752 or 83C752 Probe Card to 83C751 and 87C751 devices.

**MHW-CONV12**

A 24-lead Skinny DIP to 28-lead PLCC converter for 83C751 Probe Card to 83C751 and 87C751 devices.

**MHW-CONV13**

A 28-lead DIP to 28-lead PLCC converter for PE-83752 and 83C752 Probe Card to 83C752 and 87C752 devices.

**MHW-CONV14**

A 40-lead DIP to 44-lead PLCC converter for PE-8031, 8031 Probe Card to 8031, 80C31, 8031 and 80C31 devices.

A 40-lead DIP to 44-lead PLCC converter for PE-8032, 8032 Probe Card to 8031, 80C31, 8031, 80C31, 80C32, 8032 and 80C32 devices.

A 40-lead DIP to 44-lead PLCC converter for 8052 Probe Card to 8031, 80C31, 8031, 80C31, 80C32, 8032, 80C32, 8051, 80C51, 8751, 87C51, 8052, 80C52, 8752 and 87C52 devices.

A 40-lead DIP to 44-lead PLCC converter for 8351FB Probe Card to 8031, 80C31, 8031, 80C31, 80C32, 8032, 80C32, 8051, 80C51, 8751, 87C51, 8052, 80C52, 8752, 87C52, 8XC51FA and 8XC51FB devices.

A 40-lead DIP to 44-lead PLCC converter for 80410 Probe Card to 80CL410 devices.

A 40-lead DIP to 44-lead PLCC converter for 80652 and 83652 Probe Cards to 80C652, 87C652, 83C652, 87C654 and 83C654 devices.

A 40-lead DIP to 44-lead PLCC converter for 80528 and 83528 Probe Cards to 80C528, 87C528, 83C528, 87C524 and 83C524 devices.

A 40-lead DIP to 44-lead PLCC converter for 80851 and 83851 Probe Cards to 80C851 and 83C851 devices.

#### MHW-CONV19

A 68-lead PLCC to 44-lead PLCC converter for 83552 and 80552 Probe Cards to 8031, 8032\*, 80C31, 80C32\*, 8051, 8751, 80C51, 87C51, 8052\*, 8752\*, 80C52\*, 80C652, 83C652, 87C652, 87C654, 83C654 and 87C52\* devices.

#### MHW-CONV23

A 44-lead PLCC to 40-lead DIP converter for 80550 Probe Card to 80C550, 87C550 and 83C550 devices.

#### iceMASTER-8051 and PE Hardware Miscellaneous Products

#### MHW-EXT24DIP

A 24 Lead Skinny DIP, 6 inch target interface extension cable for 83C751 and 87C751 devices.

#### MHW-EXT28DIP

A 28 Lead DIP, 6 inch target interface extension cable for 83C752 and 87C752 devices.

#### MHW-EXT40DIP

A 40 Lead DIP, 6 inch target interface extension cable for 8031, 80C31, 8032, 80C32, 8051, 80C51, 8751, 87C51, 8051FA, 83C51FA, 87C51FA, 8051FB, 83C51FB, 87C51FB, 8052, 80C52, 8752, 87C52, 80C528, 83C528, 87C528, 83C524, 87C524, 80C550, 83C550, 87C550, 80C652, 83C652, 87C652, 83C654, 87C654, 80CL410, 80C851 and 83C851 devices.

#### MHW-EXT28PLCC

A 28 Lead PLCC, 6 inch target interface extension cable for 83C751, 87C751, 83C752 and 87C752 devices.

#### MHW-EXT44PLCC

A 44 Lead PLCC, 6 inch target interface extension cable for 8031, 80C31, 8032, 80C32, 8051, 80C51, 8751, 87C51, 8051FA, 83C51FA, 87C51FA, 8051FB, 83C51FB, 87C51FB, 8052, 80C52, 8752, 87C52, 80C528, 83C528, 87C528, 83C524, 87C524, 80C550, 83C550, 87C550, 80C652, 83C652, 87C652, 83C654, 87C654, 80CL410, 80C851 and 83C851 devices.

#### MHW-EXT68PLCC

A 68 Lead PLCC 6 inch target interface extension cable for 83C552, 87C552, 83C562, 87C562, 80C552, 83C451, 87C451, 80C451, 87C592, 80C592 and 83C592 devices.

**MHW-DIPSE24/3**

A 24 Lead Skinny DIP pin isolator for 83C751 and 87C751 devices.

**MHW-DIPSE28/6**

A 28 Lead DIP pin isolator for 83C752 and 87C752 devices.

**MHW-DIPSE40/6**

A 40 Lead DIP pin isolator for 8031, 80C31, 8032, 80C32, 8051, 80C51, 8751, 87C51, 8051FA, 83C51FA, 87C51FA, 8051FB, 83C51FB, 87C51FB, 8052, 80C52, 8752, 87C52, 80C528, 83C528, 87C528, 83C524, 87C524, 80C550 83C550, 87C550, 80C652, 83C652, 87C652, 83C654, 87C654, 80CL410, 80C851 and 83C851 devices.

**MHW-DIPSE42S/6**

A 42 Lead Shrink DIP pin isolator for 87C054, 83C054 and 83C053 devices.

**MHW-PLCSE28**

A 28 Lead PLCC pin isolator for 83C751, 87C751, 83C752 and 87C752 devices.

**MHW-PLCSE44**

A 44 Lead PLCC pin isolator for 8031, 80C31, 8032, 80C32, 8051, 80C51, 8751, 87C51, 8051FA, 83C51FA, 87C51FA, 8051FB, 83C51FB, 87C51FB, 8052, 80C52, 8752, 87C52, 80C528, 83C528, 87C528, 83C524, 87C524, 80C550 83C550, 87C550, 80C652, 83C652, 87C652, 83C654, 87C654, 80CL410, 80C851 and 83C851 devices.

**MHW-PLCSE68**

A 68 Lead PLCC pin isolator for 83C552, 87C552, 83C562, 87C562, 80C552, 83C451, 87C451, 80C451, 87C592, 80C592 and 83C592 devices.

**iceMASTER-8051 and PE Software Miscellaneous Products****MSW-ASM51**

MetaLink 8051 Macro Cross Assembler

**MSW-SIM51**

Archimedes Software SimCASE 8051 for PC-hosted systems

**MSW-SIMI/O51**

Archimedes Software Sim I/O 8051 for PC-hosted systems

**MSW-A51**

Archimedes Software Assembler 8051 for PC-hosted systems

**MSW-C51**

Archimedes Software 8051 Family C Compiler for PC-Hosted Systems

## **MSW-AVA51**

Avocet Systems, Inc. AvCase Assembler for the 8051

## **MSW-AVC51**

Avocet Systems, Inc. AvCase C-Compiler and Assembler for the 8051

## **MSW-AVS51**

Avocet Systems, Inc. AvCase Simulator for the 8051

## **MSW-BSO/A51**

BSO/Tasking Assembler 8051 for PC-hosted systems

## **MSW-BSO/C51**

BSO/Tasking C-Compiler and Assembler for 8051 for PC-hosted systems

## **MSW-BSO/PLM51**

BSO/Tasking PL/M-Compiler and Assembler for 8051 for PC-hosted systems

## **MSW-F/4010**

Franklin Software A51 Assembler for PC-Hosted Systems

## **MSW-F/5020**

Franklin Software C-51 Compiler and A51 Assembler for PC-Hosted Systems

## **MSW-F/8220**

Franklin Software 8051 Developer's Kit with C-51 Compiler, A51 Assembler, Simulator/Debugger for PC-Hosted Systems

## **MSW-F/8310**

Franklin Software 8051 Professional Developer's Kit with C-51 Compiler, A51 Assembler, Bank Linker 51, Tiny Real Time Operating System 8051, Simulator/Debugger for PC-Hosted Systems

## NOHAU EMUL51-PC – PC-based in-circuit emulator

### 1.0 System Architecture

Features of the Nohau EMUL51-PC in-circuit emulator include:

- Low-cost full real-time emulation
- IBM PC-bus plug-in boards or stand alone Box version with 115K baud RS232-C connection to IBM PC
- Easy-to-learn user interface with windows and pull-down menus
- Source-level debugging in C, PL/M or Pascal with full support for typed symbols
- 256k frames by 64 bit real-time trace option with time stamp
- Program Performance Analyzer

The Nohau EMUL51-PC consists of a board which plugs directly into the IBM PC/XT/AT bus for fast file transfer. An optional external box with a serial link is also available. The optional Trace board features an advanced trace function with many trigger capabilities.

The POD, which plugs into the target system, is connected with a 5 ft (1.5 m) ribbon cable to the Emulator board to provide a flexible operating range.

The EMUL51-PC uses no wait states and does not intrude on memory, stack, I/O or interrupt pins.

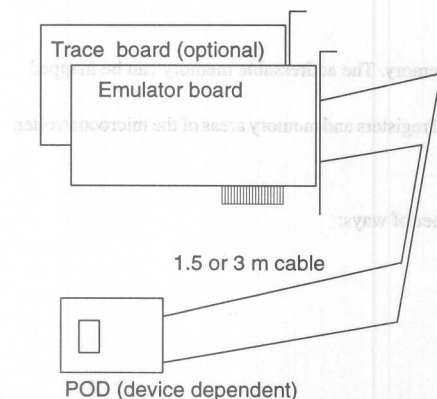


fig 1 EMUL51-PC plug-in board

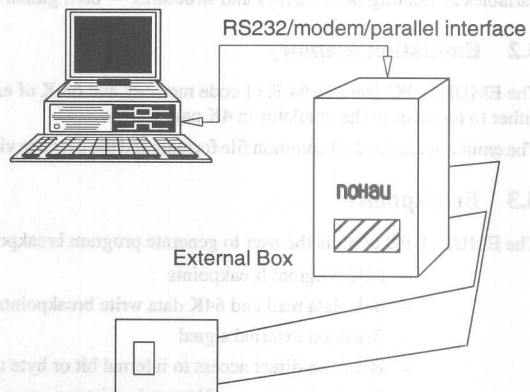


fig 2 External Box version



## 2.0 User Interface

The user interface is based on MS-DOS software. It incorporates a variety of techniques. The user can select to operate either with short typed commands or using pull-down menus. On-screen features include windows to monitor or alter:

- Assembly or high level language source code
- CPU registers
- Internal data and Special Function Registers
- External data
- Watch variables in C, PL/M or assembly
- Trace setup and display

All commands are supported with context-sensitive help and full on-line manual.

## 3.0 Specific Features

### 3.1 Source Level Debugging

Using high level language for code generation is a way to cut development time. Therefore the EMUL51-PC gives full support for debugging directly in C or PL/M source code. This eliminates the need for paper listings. Breakpoints can be marked directly in the source code window. The user can single step through the program line by line and follow the program execution on the screen listing.

The trace permits the user to trace his source code in real-time.

All variables can be displayed and altered. This includes full support for typed symbols which permits the user to address such variables as floating-point, arrays and structures — both global and local.

### 3.2 Emulation Memory

The EMUL51-PC features 64 K of code memory and 64 K of external data memory. The addressable memory can be mapped either to target or to the emulator in 4K pages.

The emulator can load all common file formats and the user can view and alter all registers and memory areas of the microcontroller.

### 3.3 Breakpoints

The EMUL51-PC permits the user to generate program breakpoints in a number of ways:

- 64K program breakpoints
- 64K data read and 64K data write breakpoints
- Break on external signal
- Break on direct access to internal bit or byte memory
- Break on contents of internal register or memory
- Break on program access out-of-boundary

Using the Trace board it is possible to break on combinations of address, data, control signals, port signals and external signals.

### 3.4 Macros and debug session logging

Test session automation is made possible using the Macro commands. This permits the user to define his own command sequences using structures like IF/ELSE and REPEAT/WHILE. Such command sequences can be stored to a file which can be loaded automatically when the emulator is invoked.

A complete debug session and all setups can also be recorded to a file.

### 3.5 Trace Memory

The optional trace board features a trace buffer capable of storing up to 256k frames of 64 bit data each. The 64 bits consist of address, data, control signals, port signals, external signals and a time stamp.

The trace memory can be displayed, reprogrammed and restarted during emulation.

#### 3.5.1 Trace Filter

The trace filter makes it possible to select what events are stored in the trace buffer. The qualifiers permit the user to define the criteria for which bus cycles are stored. The qualifiers can specify address, data, control signals, port signals and external signals.

#### 3.5.2 Trace Trigger

The trace works much like a logic analyzer. It is therefore possible to trigger the trace on an event and display what happened before or after that event.

The trigger event can be defined using any of the qualifiers in up to eight levels. It is possible to trigger on boolean combinations of the qualifiers, or sequential combinations including a loop counter.

The trigger point can be selected anywhere within the 256k trace buffer to give full choice of pre/post trigger alignment.

#### 3.5.3 Trace Display

The trace information can be displayed in high-level language statements, in disassembled form or in binary/hex form. It can also be stored to a file.

#### 3.5.4 Program Performance Analyzer

With the PPA, information can be generated showing which addresses the user program spends its time on. The information can be displayed as statistics or in histogram form.

## 4.0 General Specifications

Host:	IBM PC/XT/AT/386/486, PS/2 or compatible with 640K of RAM. Monochrome, color or enhanced graphics display.
External Box:	The emulator boards can be installed in an external box with serial communication to the PC.
Processors supported:	8051, 8052, 8031, 8032, 80C51, 80CL51, 80C52, 80CL52, 80C31, 80CL31, 80C32, 8XC053/054/055, 83/80CL410/610, 83/80C451, 8XC524, 8XC528, 8XC550, 83/80C552, 8XCE558, 80C562, 8XC575, 80CL580, 8XC592, 8XC598, 83/80C652, 83/87C750, 83/87C751, 83/87C752, 8XCL781, 8XCL782, 83/80C851 (For more details, please refer to the following pages.) A switch from one microcontroller to another is made by changing the low cost POD.
File formats supported:	Intel HEX/OBJ/SYM/OMF, Avocet, Archimedes/IAR, BSO/Tasking, Franklin/Keil, Intermetrics/Whitesmiths/Comic, and many more.
Clock speed:	Allows operation up to 33 MHz in real-time.
Power supply:	The boards are powered from the PC-bus. The Emulator board requires 1.7A/5V and the Trace board 1.3A/5V.

## 5.0 Ordering information

### PHILIPS SEMICONDUCTORS Microcontroller Support

#### EMULATOR UNITS

Includes software and ribbon cable. Must be connected to a POD to operate. Each step up in frequency rating covers all lower frequency steps. Includes DIP isolator if ordered with 40-pin POD. Emulator software has high-level debug and all options.

Order Number	Description
EMUL51-PC/E32	12 MHz Emulator, 32 kB (kiloByte) Emulation Memory.
EMUL51-PC/E32-16	16 MHz Emulator, 32 kB Emulation Memory.
EMUL51-PC/E128	12 MHz Emulator, 128 kB Emulation Memory.
EMUL51-PC/E128-16	16 MHz Emulator, 128 kB Emulation Memory.
EMUL51-PC/E128-20	20 MHz Emulator, 128 kB Emulation Memory.
EMUL51-PC/E128-24	24 MHz Emulator, 128 kB Emulation Memory.
EMUL51-PC/E128-30	30 MHz Emulator, 128 kB Emulation Memory.
EMUL51-PG/E128-33	33 MHz Emulator, 128 kB Emulation Memory.

#### TRACE BOARD OPTIONS

Optional second PC plug-in board. Emulator board contains no trace capability. Each frequency step covers all lower steps.

Order Number	Description
EMUL51-PC/TR4	12 MHz 4 kiloframe (4096 frames) Trace Buffer.
EMUL51-PC/TR4-16	16 MHz 4 k Trace Buffer.
EMUL51-PC/TR16	12 MHz 16 k Trace Buffer.
EMUL51-PC/TR16-16	16 MHz 16 k Trace Buffer.
EMUL51-PC/TR16-20	20 MHz 16 k Trace Buffer.
EMUL51-PC/TR16-24	24 MHz 16 k Trace Buffer.
EMUL51-PC/TR16-30	30 MHz 16 k Trace Buffer.
EMUL51-PC/TR16-33	33 MHz 16 k Trace Buffer.

Advanced Trace Boards feature 32-bit timestamping with 16-bit prescaler, eight-level triggers, state and counter functions, search.

Order Number	Description
EMUL51-PC/ATR64-16	16 MHz 64 k Advanced Trace Buffer.
EMUL51-PC/ATR256-16	16 MHz 256 k Advanced Trace Buffer. Contact Nohau for availability.
EMUL51-PC/ATR64-24	24 MHz 64 k Advanced Trace Buffer.
EMUL51-PC/ATR256-24	24 MHz 256 k Advanced Trace Buffer.
EMUL51-PC/ATR64-33	33 MHz 64 k Advanced Trace Buffer.
EMUL51-PC/ATR256-30	30 MHz 256 k Advanced Trace Buffer.

#### BONDOUT PODS, 24-PIN, 40-PIN, 68-PIN, 84-PIN

Allow full emulation of internal, external and mixed modes of bus or port input/output. On-board POD crystal is 12 MHz on all PODs of any frequency specification. Each step up in frequency rating covers all lower frequency steps.

Order Number	Description
POD-C51B	12 MHz Bondout POD for 80C51, 8051, 87C51, 8751, 80C31, 8031, 80C/83C652, 83C654, 80C/83C662, 80C/83C851 single-chip or external mode.
POD-C51B-16	16 MHz Bondout POD for 80C/87C51-1, 80C/87C51, 80/8751, 80C31-1, 80C31, 8031, 80C/83C652, 83C654, 80C/83C662, 80C/83C851 single-chip or external mode.
POD-C51B-24	24 MHz POD for 80C/87C51-24, 80/8751, 80C31-24, 80C/83C652/654, 80C/83C662, 80C/83C851 single-chip or external mode. <b>Special Requirement:</b> Due to bondout chip timing, this POD requires a 30 MHz emulator board and trace board must be 30 MHz if used.
POD-CL410	POD for 83CL410, 83CL610, 80CL31, 80CL51. Target voltage range: 1.5–5.0V. Maximum frequency: 12 MHz at 5.0V. This POD is intended for emulating internal code. External bus operation is limited.
POD-C451B-PGA	12 MHz Bondout POD for 83C451, 87C451, 80C451 PLCC, single-chip or external mode. PGA from POD. Use optional adapter for PLCC target.
POD-C451B-PGA-16	16 MHz Bondout POD for 83C451, 87C451, 80C451 PLCC, single-chip or external mode. PGA from POD. Use optional adapter for PLCC target.
POD-C552B-PGA	12 MHz Bondout POD for 83C552, 87C552, 80C552 PLCC, single-chip or external mode. PGA from POD. Use optional adapter for PLCC target.
POD-C552B-PGA-16	16 MHz Bondout POD for 83C552, 87C552, 80C552 16 MHz, 12 MHz single-chip or external mode. PGA from POD. Use optional adapter for PLCC target.

POD-C552B-PGA-24	24 MHz Bondout POD for 8XC552 or 8XC562 PLCC single-chip or external mode. PGA from POD. Use optional adapter (etc). <b>Special Requirement:</b> Due to bondout chip timing, this POD requires a 30MHz emulator board and trace board must be 30 MHz if used.
POD-CL580	12MHz Bondout POD for 83CL580.
POD-C652B	Order as POD-C51B.
POD-C751	12 MHz 83C750, 87C750, 83C751, 87C751, DIP POD. Includes EXT-DIP24.
POD-C751-16	16 MHz 83C750, 87C750, 83C751, 87C751, DIP POD. Includes EXT-DIP24.
POD-CL782	12 MHz POD for 83CL781, 83CL782, 83CL52

#### "HOOKS" MODE PODS, 28-PIN, 40-PIN, 42-PIN, 44-PIN, 68-PIN

Standard chip operated in special "hooks" emulation mode for single-chip or external modes. Some electrical characteristics are different from microcontroller's. On-board POD crystal is 12 MHz on all PODs of any frequency specification.

Order Number	Description
POD-C52	12 MHz POD for 80C31, 80C32, 80C51, 80C52, 87C51, 87C52.
POD-C52-16	16 MHz POD for 80C31, 80C32, 80C51, 80C52, 87C51, 87C52.
POD-C528	12 MHz POD for 80C528, 83C528, 87C528, 83C524, 83C528.
POD-C528-16	16 MHz POD for 80C528, 83C528, 87C528, 83C524, 83C528.
POD-C550-PGA	12 MHz POD for 80C550, 83C550, 87C550. PGA from POD. Use optional adapter for PLCC target.
POD-C550-PGA-16	16 MHz POD for 80C550, 83C550, 87C550. PGA from POD. Use optional adapter for PLCC target.
POD-C575	12 MHz 87C575, 87C575 chip may have to be supplied by user.
POD-C592-PGA	12 MHz POD for 8XC592. 8XC592 chip may have to be supplied by user. PGA from POD. Use optional adapter for PLCC target.
POD-C592-PGA-16	16 MHz POD for 8XC592. 8XC592 chip may have to be supplied by user. PGA from POD. Use optional adapter for PLCC target.
POD-C752	12 MHz 83C752, 87C752 DIP POD. Includes EXT-DIP28.
POD-C752-16	16 MHz 83C752, 87C752 DIP POD. Includes EXT-DIP28.
POD-C054	12 MHz POD for 8XC053, 8XC054, 8XC055 (Microcontroller-for-Television-Video).
POD-C575	16 MHz POD for 8XC575.

#### POD BOARDS, 40 PIN EXTERNAL MODE

Port 2 is upper address bus only. Port 0 is address/data bus. P3.6 is WRITE, P3.7 is READ. On-board POD crystal is 12 MHz on all PODs of any frequency specification. Each step up in frequency rating covers all lower frequency steps.

Order Number	Description
POD-31	12 MHz 8031 POD.
POD-C31	12 MHz 80C31 POD.
POD-32	12 MHz 8032 POD.
POD-C32	12 MHz 80C32 POD.
POD-C652	12 MHz 80C652 POD.
POD-C31-1	16 MHz 80C31-1 POD.
POD-C32-16	16 MHz 80C32 POD.
POD-C31-20	20 MHz 80C31 POD.
POD-C31-24	24 MHz 80C31 POD.
POD-C31-30	30 MHz 80C31 POD.
POD-C31-33	33 MHz 80C31 POD.

-S version of the above boards allows P3.6, P3.7 to be used either as unidirectional I/O or write and read. (Example: POD-31-S; POD-C31-S-1; POD-C32-S-16.)

#### EXTERNAL MODE PODS, 64-PIN, 68-PIN

Port 2 is upper address bus only. Port 0 is address/data bus. P3.6 is WRITE, P3.7 is READ. On-board POD crystal is 12 MHz on all PODs of any frequency specification. Each step up in frequency rating covers all lower frequency steps.

Order Number	Description
POD-C451-DIP	12 MHz 80C451 DIP POD.
POD-C451-DIP-16	16 MHz 80C451 DIP POD.
POD-C451-PGA	12 MHz 80C451 PLCC POD. PGA from POD.
POD-C451-PGA-16	16 MHz 80C451 PLCC POD. PGA from POD.
POD-C552-PGA	12 MHz 80C552 POD. PGA from POD. Use optional adapter for PLCC target.
POD-C552-PGA-16	16 MHz 80C552 POD. PGA from POD.
POD-C562-PGA-16	16 MHz 80C562 POD. PGA from POD.

## BOX OPTIONS

Box units are AC line-powered. Emulator software runs under DOS on PC and uses a COM port at 110 baud to 115 kilobaud. Serial cable included.

### Order Number

EMUL51-PC/BOX-S  
EMUL51-PC/BOX-CS  
EMUL51-PC/BOX-CS-20  
EMUL51-PC/BOX-CS-24  
EMUL51-PC/BOX-CS-30

### Description

Serial Box. Select emulator separately; trace optional. POD not included.  
Serial Box with E128-16 Emulator and TR16-16 Trace. POD not included.  
Serial Box with E128-20 Emulator and TR16-20 Trace. POD not included.  
Serial Box with E128-24 Emulator and TR16-24 Trace. POD not included.  
Serial Box with E128-30 Emulator and TR16-30 Trace. POD not included.

## MISCELLANEOUS OPTIONS

### Order Number

EMUL51-PC/PRG  
EMUL51-PC/PRG751  
EMUL51-PC/EXT-DIP24  
EMUL51-PC/DIP24-ISO  
EMUL51-PC/DIP24-PLCC28  
EMUL51-PC/EXT-DIP24-PLCC28  
EMUL51-PC/EXT-DIP28  
EMUL51-PC/DIP28-PLCC28  
EMUL51-PC/DIP28-ISO  
EMUL51-PC/DIP28-DIP24-ADAP

### Description

Universal Programmer (EPROM, 8751, 87C51, PAL).  
Programmer for 87C751, 87C752.  
Additional extender cable for 24 pin DIP.  
24 pin DIP Isolator.  
24 pin DIP to 28 pin PLCC.  
Extender cable, 24 pin DIP to 28 pin PLCC.  
Extender cable, 28 pin DIP.  
28 pin DIP to 28 pin PLCC.  
Additional 28 pin DIP Isolator.  
28-pin (0.600-In) to 24-pin (0.300-In) adapter to plug POD-C752 into 8XC751 target. The user is responsible for port and register compatibility.

EMUL51-PC/EXT-DIP40  
EMUL51-PC/DIP40-ISO  
EMUL51-PC/DIP40-PLCC44  
EMUL51-PC/DIP40-ONCE-DIP40

Extender cable for 40 pin DIP.  
Additional 40 pin DIP Isolator.  
40 pin DIP to 44 pin PLCC.  
Clips over target microcontroller. Disables target micro to allow emulation without removing target chip. Works only on chips with on-chip emulation (ONCE) disable feature.

EMUL51-PC/DIP40-ONCE-PLCC44

Clips over target microcontroller. Disables target micro to allow emulation without removing target chip. Works only on chips with on-chip emulation (ONCE) disable feature.

EMUL51-PC/PGA44-PLCC44  
EMUL51-PC/PGA44-PLCC44-EL2  
EMUL51-PC/EXT-DIP48  
EMUL51-PC/DIP48-ISO  
EMUL51-PC/PGA68-PLCC68  
EMUL51-PC/PGA68-DIP84  
EMUL51-PC/PGA68-ISO  
QILEXT-1  
EMUL51-PC/EZ

PGA to PLCC adapter, 44 pin.  
PGA to PLCC "elevator" or "tower" rigid 2-inch extender-adapter.  
Extender cable for 48 pin DIP.  
48 pin DIP Isolator.  
PGA to PLCC adapter unit, 68 pin.  
PGA to DIP adapter unit for 451 PGA PODs to plug into DIP target.  
68 pin PGA Isolator.  
Extractor tool for PLCC parts.  
E-Z-Hook® wires for trace.

E-Z-Hook is a registered trademark of Tektest, Inc.

EMUL51-PC/CBL10-S  
EMUL51-PC/CBL10-A  
EMUL51-PC/CBL5-A

10 foot substitute for 5 foot POD cable (not for bondout PODs).  
Additional 10 foot POD cable (not for bondout PODs)  
Replacement 5 foot POD cable

## BANKSWITCH EMULATOR BOARDS

User selectable as two banks of 64 kBytes, or as three switchable banks in upper half (8000 – FFFF) with lower half (0000 – 7FFF) not switchable. MOVX read-write data memory is only separate from code if data is mapped to target.

### Order Number

EMUL51-PC/E128-BSW  
EMUL51-PC/E128-BSW-16  
EMUL51-PC/E256-BSW  
EMUL51-PC/E256-BSW-16  
-BSW option to any POD

### Description

12 MHz Bankswitch Emulator, 128 kB Emulation Memory. Requires Bankswitch POD.  
16 MHz Bankswitch Emulator, 128 kB Emulation Memory. Requires Bankswitch POD.  
12 MHz Bankswitch Emulator, 256 kB Emulation Memory. Requires Bankswitch POD.  
16 MHz Bankswitch Emulator, 256 kB Emulation Memory. Requires Bankswitch POD.

### Examples:

POD-31-BSW  
POD-C31-BSW-1  
POD-31-S-BSW

12 MHz 8031 Bankswitch POD.  
16 MHz 80C31-1 Bankswitch POD.  
12 MHz 8031 -S option Bankswitch POD.



## SOFTWARE PACKAGES

### Order Number Description

#### NOHAU Corporation

##### SIMUL51-PC

##### EMUL51-PC/SWSUB

##### Archimedes Software, Inc.

##### ARCHM/C-8051PC

##### ARCHM/SIM-8051PC

Nohau 8031/8051 Simulator. Same interface as EMUL51-PC.

EMUL51-PC emulator software update service, one year.

Archimedes C-8051PC V4 C-Compiler & Assembler.

Archimedes 8051 Simulator/Debugger (SimCASE Sim-8051PC) for CCS1, PL/M-51 & ASM.

Archimedes is a trademark of Archimedes Software, Inc.

#### Avocet Systems, Inc.

##### AVOCET/AVA51

##### Avocet Systems AvCase51 8051 Assembler

Avocet is a registered trademark of Avocet Systems, Inc.

#### BSO/Tasking

##### BSOTSK/C51PKG

##### BSOTSK/PLM51PKG

##### Chip Tools, Inc.

##### CV51-NOH

##### CV51-S

##### CV51-SNOH

BSO/Tasking 8051 Family C-Compiler and Assembler Package.

BSO/Tasking 8051 Family PL/M Compiler and Assembler Package.

Chip Tools Chip View-51 High-Level/Low-Level Debugger Emulator interface for EMUL51-PC.

Chip Tools Chip View-51 High-Speed Simulator.

Chip Tools Chip View-51 CV51-NOH + CV51-S Combo package.

Chip Tools and Chip View are trademarks of Chip Tools, Inc.

#### Franklin Software, Inc.

##### FRANKLIN/4010

##### FRANKLIN/5020

##### FRANKLIN/7020

##### FRANKLIN/8220

Franklin 8051 Macro Assembler with Linker, Librarian, Object-to-Hex utility.

Franklin V3 Very High Performance 8051 Compiler & Assembler.

Franklin V5 Simulator/Debugger, Windowed Interface.

Franklin Developers Kit with 5020 Compiler, 4010 Assembler and 7020 Simulator/Debugger.

Franklin is a trademark of Franklin Software, Inc.

#### Intermetrics Microsystems Software, Inc./Whitesmiths, Ltd.

##### INTERMET/C851HX

##### INTERMET/D851HXNOH

##### INTERMET/D851HXSIM

Intermetrics Whitesmiths C-Compiler and Assembler, Extended.

Intermetrics Whitesmiths C Source-Level Debugger/Emulator Version (Interface for EMUL51-PC).

Intermetrics Whitesmiths C Source-Level Debugger/Simulator Version.

Whitesmiths and CXDB are registered trademarks of Intermetrics, Inc.

The EMUL51-PC Emulator, Trace, POD, and Box hardware is sold with a one-year warranty. The EMUL51-PC Emulation software is sold with no warranty, but upgrades will be distributed to all customers up to one year from the date of purchase. Nohau Corporation makes no warranties, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. In no event will Nohau Corporation be liable for consequential damages. The SIMUL51-PC Simulator software includes updates for a period of one year. Third-party software and programmers sold by Nohau carry manufacturers' warranties.

## CHOOSING PODS for the EMUL51-PC

People sometimes ask: "How do I choose which POD to use for the 8051-family chip I want to emulate?"

Which POD you pick for your application depends on several things. What micro you are using, the frequency, the mode you are using it in, your target configuration, and price all play a part. The **EMUL51-PC ORDER INFORMATION** can help you choose among the POD types available from Nohau.

Nohau has several categories of PODs: (1) External-mode PODs; (2) Bondout PODs; and (3) "Hooks"-mode PODs.

### 1. External-Mode PODs

External or microprocessor-mode PODs have several clear advantages. They are the cheapest type of POD. They have easily-replaceable standard microcontrollers. Your program runs on the real production part, with most of the lines directly connected to your target. You can use them where your microcontroller has external program or external read-write memory. With this type of POD, Port 2 is used only for the upper address bus and not as port input-output (I/O). Port 0 is used as a multiplexed address and data bus. In most of these PODs, P3.6 can only be used as the write line and P3.7 as only the read line. The **POD-31** is a typical external POD.

Your target system shares Port 0 and Port 2 with the emulator. The emulator uses Port 0 to run its monitor code when it is not executing your code in real-time. When you are not running real-time emulation, the POD holds the Program Store ENable line (PSEN/) high so that your external read-only memory (ROM) is not enabled. It also holds the Read/ line and Write/ lines high so that none of your external read-write random access memory (RAM) or I/O is enabled.

But in the monitor (not emulating) mode, the address lines of both Port 2 and Port 0 are active and can output any address in the 64 kByte address range. It is up to your target system to prevent enabling any device unless the PSEN/ or the RD/ or the WR/ line goes low.

If you are emulating a ROM-less part, like the 8031, you can use the **POD-31** type of POD. You also might use this POD for some internal ROM applications. You could use it for designs that have all of Port 2 and Port 0 used as external buses. This is true even though the program can be executed from on-chip 8051 program memory in your final product. This is a case where your target schematic, rather than the device you are using lets you use this type of POD. So though the part may ultimately be an 8051 or 8751, you can use a **POD-31** because you are using the ports as buses.

If you are emulating the 80C31, 8032, 80C32 or 80C652, you can get external-mode PODs for these microcontrollers. With the **POD-31**, you can emulate all those parts in external mode at up to 12 MHz. You can get the POD with the correct micro installed, such as a **POD-32**. Or you can change among the types by changing the microcontroller component in the POD yourself.

16 MHz, 20 MHz, 24 MHz, 30 MHz and 33 MHz PODs are available for 40-pin parts. If you use a **POD-C31-1** with a 16 MHz rated emulator, you can support such parts as the 80C31-1, 80C32-1 and 80C652-1. Any higher-frequency POD and emulator set can support all the lower frequencies.

For emulating the 80C451, 80C552, or 80C562 other external mode PODs are available. PODs for dual inline package (DIP) parts have a DIP plug coming out of the bottom of the POD. PODs for plastic leaded chip carrier (PLCC) parts have a pin-grid array (PGA) plug coming out from the bottom of the POD. To plug a PGA POD into a PLCC socket, you can get an optional adapter. The PGA68-PLCC68 is a typical adapter. If your target board has feed-through holes and you solder a PGA socket into it, you don't need an adapter. If your target board already has a PGA socket, you don't need an adapter.

**POD-31-S:** All the above external PODs use P3.6 as WR/ and P3.7 as RD/. But there is a 40-pin external mode POD version that lets you use these two lines as I/O. The basic board is the **POD-31-S**. This special -S option lets you select whether the two lines are to be WR/ and RD/ or else I/O. They can be unidirectional port lines (input-input, input-output, output-input, or output-output). You also can get this POD in the variations of 16 MHz and 20 MHz and any of the 40-pin part variations listed above.

## 2. Bondout PODs

Bondout PODs use special microcontrollers that the semiconductor manufacturers designed to perform all the functions of the part. They also have additional lines "bonded out" from the chip that allow control for emulation. PODs with these special chips cost more than PODs with widely available commercial chips, but give you a greater flexibility in how you use the ports. These PODs allow you to use any combination of internal or external mode of the chip, and respond to the state of the External Access (EA) pin. Nohau has bondout PODs for many different microcontrollers.

You can emulate the 80/80C51, 87/87C51, 80/80C31, 80C/83C/87C652, 83C/87C654, 80C/83C662 and 80C/83C851 with the **POD-C51B**. 16 MHz and 24 MHz versions are available.

To emulate the 83C451, 87C451 and 80C451 in the PLCC package, you can use the **POD-C451B-PGA**. See the previous discussion about PGAs.

For emulating the 80C/83C/87C552 and 80C/83C/87C562 you can use the **POD-C552B-PGA**. It also is available in a 16 MHz and 24MHz versions.

For emulating the 83C750, 87C750, 83C751 and 87C751, use the **POD-C751**. It has a 24 pin DIP plug on the bottom of the POD. You also can get an adapter to plug into 28 pin PLCC sockets. A 16 MHz version is available.

For emulating the 8XCL410, 8XCL31, and 8XCL51, use the **POD-CL410**.

For emulating the 83CL580, use the **POD-CL580** and a special adapter.

For emulating the 83CL781, 87CL781, 83CL782, 87CL782 and 80CL52, use the **POD-CL782**.

You should be aware of one small problem that most 8051 bondout parts have with serial communication. Specifically, if your program has written to SBUF, but the TI flag has not been set, and you then break emulation, the TI flag will never be set after you resume emulation.

Bondout PODs give you the best of both worlds. They let you mix how you use the ports. So you can emulate the single-chip mode using the ports as I/O. Or you can emulate external bus mode. Or you can emulate a mix of both modes.

## 3. "Hooks"-Mode PODs

The newest type of POD Nohau offers uses a different technique for emulating. Certain chips can be put into a proprietary "hooks" mode that multiplexes the port information in and out of the part. The chip can still put out address and data. Most have 16 MHz versions available. This group includes these PODs:

The **POD-C752** is for emulating the 83C752 and 87C752. It has a 28 pin DIP plug on the bottom of the POD. You also can get an adapter to plug into 28 pin PLCC sockets.

The **POD-C52** can emulate the 80/80C/87/87C51, 80/80C/87/87C52, 80/80C31 and 80/80C32. You can use this POD when you need to emulate the 8052-type parts in single-chip mode when you are using Timer 2.

The **POD-C528** can emulate the 80/80C/87/87C528 and 83/87C524.

The **POD-C592-PGA** can emulate the 8XC592. See the previous discussion about PGAs.

The **POD-C550-PGA** can emulate the 80/80C/87/87C550. It has a 44-pin PGA plug. You also can get an adapter to plug into 44-pin PLCC sockets.

The **POD-C558-16** can emulate the 83CE558 and 89CE558.

The **POD-C575** can emulate the 8X575.

The **POD-C054** can emulate the 8XC053/054/055 "MTV" chip.

With the "Hooks"-Mode PODs, you use jumpers to select whether the code is to be fetched from the emulator RAM or from external ROM. You also select whether the read-write memory is in the emulator or on your target board. The ports on these PODs may have slightly different electrical characteristics than the

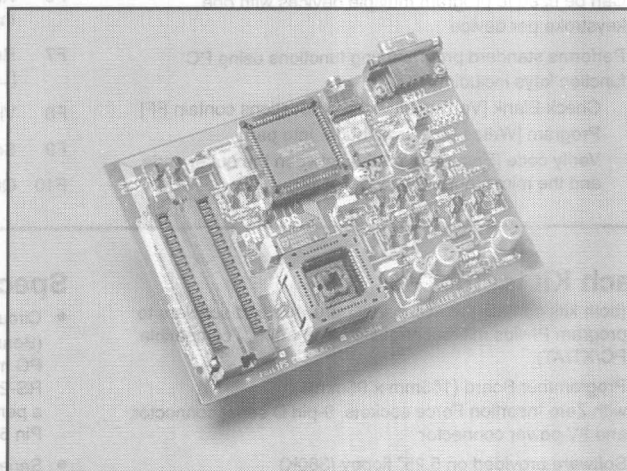
You can use any size EL

# LCP

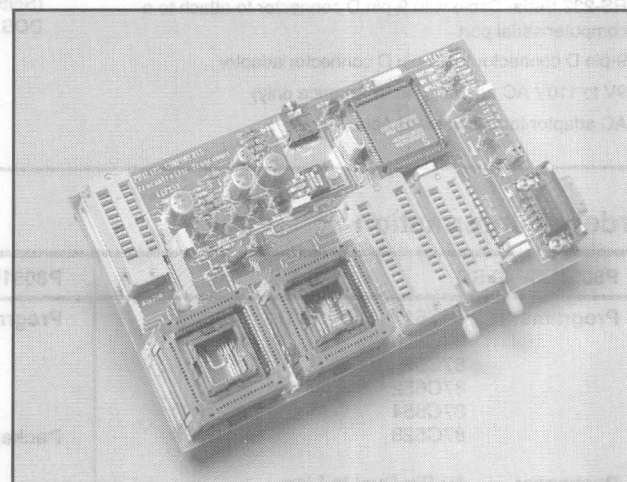
## 80C51 Family Programmiers

The LCP family of 80C51 programmers supports the entire line of Philips 80C51 microcontroller derivatives. The LCP is a low cost approach to programming microcontrollers. It was designed to be used with a standard PC. The PC provides the "brains" or processing power and buffer memory. The LCP programmers are ideal for development purposes or short production runs. There are two models of the LCP programmer; one for the standard 40-pin DIP and 44-pin LCC devices, and the other for 24- and 28-pin DIP and 68-pin LCC devices.

- Low cost programmers for 80C51 family of microcontrollers
- Serially linked to IBM PC—program time is approximately 3 minutes for 32K bytes
- Programs and verifies code bytes and security bits
- Programs microcontroller from standard hexadecimal files from PC
- Provides one button function key routines for common operations
- Allows programming multiple devices with the same code using a single command to repeat the blank check/program/verify function



**P8051LCP40 SD Programs 40-Pin DIP and 44-Pin LCC Devices**



**P8051LCPX SD Programs 24- and 28-Pin DIP and 68-Pin LCC Devices**



# Microcontroller Programmers

## Description

- Serially linked to IBM compatible host by RS-232 serial link
- Programs and checks security bits
- Reads and verifies microcontroller code
- Can be used to program multiple devices with one keystroke per device
- Performs standard programming functions using PC function keys including:
  - F1 Check Blank [Verifies all memory locations contain FF]
  - F2 Program [Writes code from buffer into part]
  - F3 Verify code [Ensures a match between the buffer code and the microcontroller code]
  - F4 Verify Security bit [Checks status of security bit]
  - F5 Auto Program (for multiple devices) [Allows sequential programming of several microcontrollers]
  - F6 Read File (load buffer) [Reads a specified HEX file into the buffer from disk]
  - F7 Read ROM (reads microcontroller ROM is not secured) [Loads code in microcontroller into PC buffer]
  - F8 View buffer [Allows user to view code]
  - F9 Setup [Calls the setup menu to select parameters]
  - F10 Quit [Returns to programmer main menu or to DOS]

## Each Kit Includes:

(both kits contain the necessary hardware and software to program Philips microcontrollers using an IBM compatible PC/XT/AT)

- Programmer Board (165mm × 95mm) with Zero Insertion Force sockets, 9-pin D serial connector, and 9V power connector
- Software provided on 5.25" floppy (360K)
- User Manual with step-by-step instructions
- RS-232 Serial Cable with 9-pin D connector to attach to a computer serial port
- 9-pin D connector to 25-pin D connector adaptor
- 9V to 110V AC adaptor (North America only)
- AC adaptor for Europe and Asia

## Specifications:

- Circuit board—with Zero Insertion Force sockets (Board does not contain program buffer—buffer is kept in PC memory)  
RS-232 communication port for serial communications with a personal computer [Pin 2 RXD/TXD, Pin 3 TXD/RXD, and Pin 5 is Ground] (Jumper block on board)
- Serial Cable with D connectors (approximately 6 feet long)
- Programming software and manual (Software only runs on an IBM compatible PC/XT/AT under DOS 2.0 or higher)

## Ordering Information

P8051LCP40 SD	P8051LCPX SD
<b>Programs:</b> 87C51 87C550 87C52 87C652 87C654 87C528	<b>Programs:</b> 87C751 (DIP) 87C752 (DIP) 87C451 (PLCC) 87C552 (PLCC)
<b>Packages:</b> 40-Pin Dual In-Line 44-Pin PLCC	<b>Packages:</b> 24-Pin Dual In-Line 28-Pin Dual In-Line 68-Pin LCC 68-Pin LCC (alternate pinout)
Available through Philips distributors.	Available through Philips distributors.

NOTE: Adaptor sockets are available from third parties for other packages.

**For more information, call: (800)447-1500  
for the number of your nearest Philips Semiconductors Sales Office.**

## Development System for 80C51 and Derivatives

PDS51

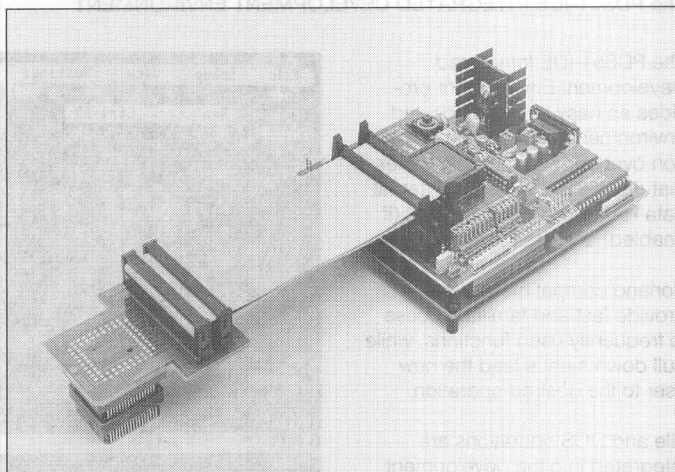
### INTRODUCTION

The PDS51 is a board-level, full featured, In-Circuit Emulator for the Philips 80C51 family of Microcontrollers. It allows the user complete access to the internal registers and full execution control with no chip resources being consumed. As standard, the PDS51 is supplied with sufficient memory to enable emulation of the complete 64K of directly addressable code memory and a 28K line by 32 bit wide, real-time trace buffer.

The PDS51 system consists of two interconnected modules. One board (the motherboard), contains the systems that are common to any derivative emulation such as the control microcontroller, logic, communication interface and power supply. The second board (the daughterboard), contains the bondout microcontroller which determines the target devices able to be emulated. Most daughter boards are able to emulate more than one target, making the range of devices available for emulation broad with minimal expense.

External breakpoints, trace control, emulation running and fetch address controlled output trigger signals are provided for interfacing and synchronising to external equipment.

The PDS51 is controlled via an RS232 serial interface by a PC running terminal emulation or the PDS51-IDE debugger supplied. The Integrated Development Environment is a windowed command and display environment which runs on PC-XT, AT, 386, 486 or compatibles.



PDS51 pictured with PDB552

### FEATURES

- Supports Philips 80C51 family
- Universal motherboard
- Bondout dependant Daughterboard
- In-Circuit emulation
- No stolen resources
- 64K emulation code space
- Hardware Breakpoints (to 64K) on:
  - Fetch address
  - Fetch address and external signal
- Real-time trace (28K steps by 32 bits)
- Traced information
  - Fetch address
  - Port pins
- External trace probes
  - Control info (inta, c1, movx)
- Conditional trace on:
  - Fetch address
  - External signal
- RS232 interface to PC
- Controlled via Terminal Emulation with SDS command set or Debugger
- Integrated Development Environment:
  - Window interface
  - Pull down menus
  - Symbolic or Source level debug
  - Borland style key strokes
  - File interface via Hex or OMF51
- Low cost

## Development System for 80C51 and Derivatives

PDS51

### The PDS51-IDE INTEGRATED DEVELOPMENT ENVIRONMENT

The PDS51-IDE Integrated Development Environment provides an easy to use windowed environment that consists of five non-overlapping resizable panes that display Code, SFR's, internal data RAM, external data RAM (if enabled) and the stack area.

Borland compatible hot-keys provide fast and familiar access to frequently used functions, while pull down menus lead the new user to the desired operation.

File and DOS operations are integrated into the development environment with hot-key (or menu selected) entry to edit, assemble and load.

File	Debug	View	Settings	Options	8xC654 Stopped
<b>Code</b>				<b>SFR's</b>	<b>Stack</b>
0000 01	Run	Ctrl-F9		ACC	01h PSU 0000000b 12:12
forgrnd:	HALT	Ctrl-Bk		R0 (0)	00h @R0int 00h 11:11
0002 B2				R1 (0)	10h @R1int 10h 0F:0F
0004 80	Reset	Ctrl-F2		P2	11110000b P0 00000000b 0D:0D
0006 00				P3	'A' P1 11111111b 0C:0C
0007 00	Run to Cursor	F4		EA	1 P2.4 1 0A:0A
0008 00	Step Into	F7		ET0	1 P3.5 0 09:09
0009 00	Step Past	F8		TL0	07h 07:07
000A 00					
000B D8	GoTo	Ctrl-G			
000D 78 20	nov	r0, #p35_rate			
000F B2 B5	cpl	p3.5			
0011 15 90	dec	p1			
0013 07	inc	@r1			
0014 32	retl				
0015 E4	clr	a			
0016 78 FF	nov	r0, #FF			
0018 E8	nov	a, r0			
0019 F6	nov	@r0, a			

File	Debug	View	Settings	Options	8xC654 Stopped
<b>Code</b>				<b>SFR's</b>	<b>Stack</b>
0000 01 15	a jmp	cstrt	R0=Z1	ACC	00h PSU 00000000b 12:12
forgrnd:				R0 (0)	21h @R0int 21h 10:10
0002 B2 90	cpl	p0.0		R1 (0)	10h @R1int 10h 0F:0F
0004 80 FC	s jmp	forgrnd		P2	11111111b P0 11100000b 0D:0D
0006 00	nop			P3	'A' P1 11111111b 0C:0C
0007 00	nop			EA	0 P2.4 1 0A:0A
0008 00	nop			ET0	0 P3.5 0 09:09
0009 00	nop			TL0	00h 07:07
000A 00	nop				
000B D8 07	d jnz	r0, nxtint			
000D 78 20	nov	r0, #p35_rate			
000F B2 B5	cpl	p3.5			
0011 15 90	dec	p1			
0013 07	inc	@r1			
0014 32	retl				
0015 E4	clr	a			
0016 78 FF	nov	r0, #FF			
0018 E8	nov	a, r0			
0019 F6	nov	@r0, a			

The code pane displays disassembled symbolic source or a full source listing and a cursor to direct operations. SFR and memory operands under the cursor are evaluated and displayed at the top of the pane.

Breakpoints, triggers, trace inhibits and external break points may be hot-key set and cleared at the cursor while remaining visible to the user as background shading of the source.

The SFR pane layout, content and display format are user defined and may be edited at any time. Valid display formats include hex, ascii, binary, signed and unsigned decimal.

## Development System for 80C51 and Derivatives

PDS51

Data entry to SFR, internal and external memory may be made in hex, ascii, octal, decimal or binary at user discretion, reinforcing the fast and friendly interface concept.

The huge 28K x 32 bit trace buffer is loaded in real time to provide subsequent viewing of past events. Executed code is presented in symbolic format on a cycle by cycle basis along with 13 user selectable sampled inputs. Trace buffer viewing is aided by a cursor which is execution time referenced to the current program counter. Additional windows offer an execution profiler for identifying hot-spots for subsequent code optimization.

File	Debug	View	Settings	Options	8x0654 Stopped
Code = B					
SFR's					
Trace Buffer					
addr	instruction	status	TR0	TR1	Stack
0018	mov a, r0	fetch	11111111	01111111	110
0019	mov r0, a	fetch	11111111	01111111	0F
001A	djnz r0, 0018	fetch	11111111	01111111	0E
001B		dummy	11111111	01111111	0D
001C	setb ea	fetch	11111111	01111111	0C
001E	setb et0	fetch	11111111	01111111	0B
0020	setb tr0	fetch	11111111	01111111	0A
0022	mov r0, #p35_rate	fetch	11111111	01111111	09
0024	mov r1, #10	fetch	11111111	01111111	08
0026	mov p2, #F0	fetch	11111111	01111111	07
0028		dummy	11111111	01111111	
0029	sjmp forgn	fetch	11110000	01111111	20
002A		dummy	11110000	01111111	1F
0002	cpl p0.0	fetch	11110000	01111111	1E
0004	sjmp forgn	fetch	11110000	01111111	1D
0005		dummy	11110000	01111111	1C
delta = -9 cycles (-9.000 us)					
0018 E8	mov a, r0	0000	00 00 00 00 00 02 00 00	00000000	
0019 F6	mov r0, a	0000	00 00 00 00 00 02 00 02	00000000	

### TERMINAL EMULATION COMMAND SET

#### Power-Up Commands

RESET Back to initialization

#### Program Execution Commands

BRA Break on address, addresses or ranges of addresses

BRn Break at given address (n = 0, 1, 2, 3)

BRR Break within given address range

BRB Break at branch instruction

BRX Break on address, addresses or ranges of addresses under external event control

BV Break on internal RAM value

GO Initiates program execution (specified addresses)

(FROM, TILL)

STEP (FROM) Executes single instruction or instructions (from a

FROM)

UD

specified address)

User-defined memory address on display

#### Memory Access Commands

DBYTE Displays specified byte from internal data memory

XBYTE Displays specified byte from external data memory

CBYTE Displays specified byte from code memory

RBYTE Displays specified byte from on-chip register memory

RBIT Displays specified bit from on-chip bit-addressable memory

ASM Assemble single instruction mnemonic into program memory

DASM Disassemble memory values into mnemonics

#### Memory Set Commands

All registers can be set and displayed by commands equal to their names (PSW, SCON, P3, TH1, TL1, TLO, etc...)

#### Serial I/O Interace Commands

SAVE Copies data from PDS51 program memory to host disk file (hex)

LOAD Transfers file (hex) from host to PDS51

#### Trace & Event Control Commands

TRACE Display executed instruction flow as sampled in real-time

TRIGGER Address, addresses or ranges of fetch addresses which control the external trigger o/p

TRACE INHIBIT Address, addresses or ranges of fetch addresses which can be excluded from tracing



## Development System for 80C51 and Derivatives

Development System  
for 80C51 and Derivatives  
**PDS51**

### SPECIFICATIONS

#### Emulation:

- 64K bytes of code space
- in-circuit emulation via bondout
- no stolen resources

#### Trace Memory:

- Real-time sampling
- 28K bytes deep 32 bits wide
- 16 bits address
- 13 bits of port lines or user signals via test clips (sampled S5P1)
- 3 bits of control (C1, inta, movx)
- input filtering on fetch address

#### Break Points:

- Hardware based
- up to 64K on fetch address
- up to additional 64K on external signal and fetch address

#### User Interfaces:

- (1) Dumb Terminal Emulation (not supplied)  
Command Line Instructions
- (2) Windowed full screen debug environment. Requires:  
PC XT/AT/386/486 or compatible  
MSDOS ver 3.10 or later  
512K bytes RAM or greater  
Monochrome/EGA/VGA Screen interface

#### Interface Signals:

- (TTL, 47K pullup terminated)
- FT Force Trace I/P  
override trace input filter
- BP Breakpoint I/P  
external event breakpoint

- Fetch address sensitive  
(Sampled S2P1)
- EM Emulation O/P  
signals Emulation running
- ET External Trigger O/P  
Fetch address controlled  
trigger O/P  
64K resolution (updated S2P1)

#### File Format:

Intel Hex, OMF51

#### Serial Interface:

- RS232C
- Asynchronous, 8 data, no parity, 1 stop bit
- 9 Pin 'D' connector
- Rxd/Txd pin 2, 3 jumper swappable
- 9.6K/19.2K/38.4K/115.2K Baud jumper selectable
- Xon/Xoff handshaking

#### Size:

Motherboard and daughterboard  
130mm x 110mm x 42mm (L.W.H.)

#### Speed:

- 3.5 to 12 MHz
- no wait states
- no stolen cycles

#### Power Supply:

- 9V nom (6.5 to 9.5V)
- 800mA max, 500mA typ (PDS51 + PDB654)
- 3.5mm Phono Plug centre positive connection

### ORDER CODES

**PDS51** Mother board complete with 64K of Emulation memory. Requires Daughter Board for operation

**PDB654** Daughter Board incorporating 654 bondout. With PDS51 emulates 87C654/652/51 Supplied with 40pin DIL footprint

**PDB52** Daughter Board incorporating 52 bondout. With PDS51 emulates 87C52/51 Supplied with 40pin DIL footprint

**PDB752** Daughter Board incorporating 752 bondout. With PDS51 emulates 87C752/751/750 Supplied with 28pin DIL and 24pin DIL footprints

**PDB528** Daughter Board incorporating 528 bondout. With PDS51 emulates 87C528/524 Supplied with 40pin DIL footprint

**PDB552** Daughter Board incorporating 552 bondout. With PDS51 emulates 87C552 / 562 / 654 / 652 / 51 Supplied with 68pin PLCC footprint and 40pin DIL footprint

**PDA44** Footprint adaptor DIL40 to PLCC44



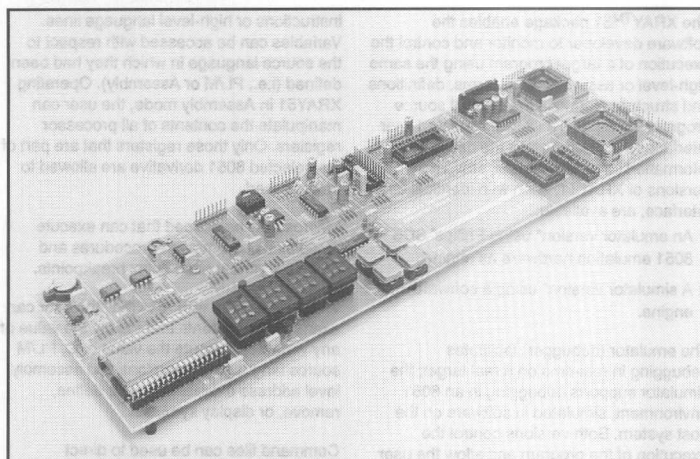
80C51 - I<sup>2</sup>C Bus Evaluation Board

## S87C00KSD

The S87C00KSD Evaluation Board is a low cost, stand-alone, board level product, designed for evaluation of the I<sup>2</sup>C Bus and Philips 80C51 derivative microcontrollers. Its modular board layout and modular software make it useful for fast prototyping and as an educational tool.

This attractive board is supplied with an 87C751 (OTP) microcontroller programmed with the demonstration software. Sockets for the 87C752, 87C654/528 and the 87C552 are included on the board. They are provided to facilitate customer development of software for these microcontrollers.

A comprehensive user manual is supplied which includes circuits, layouts, and demonstration source code listing. The user needs only to provide power from a 9V 200mA plug pack or similar DC power supply. A 3.5mm socket for this is provided on the board.



The S87C00KSD has 16 sections demonstrating and displaying 10 I<sup>2</sup>C-bus devices.

## SECTION

- Microcontroller
- LCD Display
- LED Display
- 8-bit Parallel I/O
- 8-bit Parallel I/O
- RAM
- EEPROM
- Clock / Calendar
- DTMF Dialler
- A/D and D/A
- Socket for 87C752
- Socket for 87C654 / 528
- Socket for 87C552 with external RAM and latch
- RS232 Interface
- Power Supply
- Prototyping Area

## DEVICE

- 87C751
- PCF8577 / LTD226F-12
- SAA1064
- PCF8574A
- PCF8574
- PCF8570
- PCF8582
- PCF8583
- PCF3312 / TDA7050T
- PCF8591

## Symbolic debugging package XRAY51 for the SDS 8051 emulator

OM4129

The XRAY<sup>TM</sup>51 package enables the software developer to monitor and control the execution of a target program using the same high-level or assembly-level terms, definitions and structures found in the original source program. It employs a window-oriented user interface which segregates the debugging information into meaningful areas. Two versions of XRAY51, each with identical user interface, are available:

- An emulator version\* using Philips' SDS 8051 emulation hardware as engine
- A simulator version\* using a software engine.

The emulator (debugger) facilitates debugging in real-time on a real target; the simulator supports debugging in an 8051 environment, simulated in software on the host system. Both versions control the execution of the program and allow the user to stop, start, and examine the target software by means of a powerful command language that employs C language expressions.

### XRAY51 HIGH-LEVEL DEBUGGER

The XRAY51 debugger helps to locate programming errors in the source code of C, PL/M or assembly language programs for the 8051 family of microcontrollers. The user can isolate these errors by controlling and monitoring program execution. For example, the user can single-step through the program a specified number of microcontroller

instructions or high-level language lines. Variables can be accessed with respect to the source language in which they had been defined (i.e., PL/M or Assembly). Operating XRAY51 in Assembly mode, the user can manipulate the contents of all processor registers. Only those registers that are part of the selected 8051 derivative are allowed to be accessed.

Macros may be defined that can execute complex user command procedures and provide a variety of complex breakpoints.

When debugging with XRAY51, the user can examine the contents and modify the value of any variable, compute the value of C, PL/M source language expressions and assembly level address expressions, and define, remove, or display symbols.

Command files can be used to direct XRAY51 to read or write simulated microprocessor input/output from or to a file, allowing easy implementation of automated test sequences. Command files enable scripts of debugger commands to be processed automatically without the need of user interaction.

XRAY51 also allows a virtually unlimited number of user-defined windows.

### FEATURES

The XRAY51 debugger has been designed with the developer of embedded applications in mind. Its principal features are:

- Integrated PL/M-51 source-language and assembly-language debugging; toggle by function key at any time
- Window-oriented display which segregates debugging information into meaningful areas
- Symbolic debugging with C, PL/M-51 variables and C, PL/M-51 statements
- Simple and complex breakpoints
- Single-step execution
- User-definable screens and viewports—ability to write selected information
- Command macros
- Breakpoint macros (limited to the number of breakpoints of the SDS 8051)
- Command breakpoint macros may contain C statements, including: FOR, WHILE, DO, PRINTF, and FPRINTF
- On-line context-sensitive help
- Command files
- Output logging
- Fully supports SDS 8051 with the latest firmware version
- High-level trace
- Transparent ICE mode for direct control of SDS 8051
- Supports most 8051-derived microcontrollers, including the 80C51/31, 87C51, 8XC451/528/552/562/592/652/654/751/851, 8XCL410/710/51 and the 8051/31/52/32.

<sup>TM</sup> XRAY is a trademark of Microtec Research Inc., which holds all intellectual property rights for XRAY51.

\* The SDS 8051 emulator version of XRAY51 is available from Philips Semiconductors (type number OM4129) and is sourced from BSO/Tasking Software B.V., Amersfoort, The Netherlands. The simulator version is available from BSO/Tasking Software B.V.

# Symbolic debugging package XRAY51 for the SDS 8051 emulator

OM4129

## COMMANDS

Note, all commands can be issued in an abbreviated form.

### Session Control Commands

HOST	Enter the host operating system environment
LOAD	Load an object module for debugging
QUIT	Terminate a debugging session

### Execution and Breakpoint Commands

BREAKINSTRUCTION	Set an instruction breakpoint
CLEAR	Clear a breakpoint
GO	Start or continue program execution
GOSTEP	Execute macro after each instruction step
STEP	Execute a specified number of instruction lines
STEPOVER	Step, but execute through procedures

### Display Commands

DISASSEMBLE	Display disassembled memory (assembly mode)
DUMP	Display memory contents
EXPAND	Display all local variables of a procedure
FIND	Search for a string
FOPEN	Open a file or device for writing
FPRINTF	Print formatted output to a viewport or file
LIST	Display source code
MONITOR	Monitor variables
NEXT	Find next occurrence of a string
NOMONITOR	Discontinue monitoring variables
PRINTF	Print formatted output to command viewport
PRINTVALUE	Print the value of a variable

### Memory Commands

COMPARE	Compare two blocks of memory
COPY	Copy a memory block
FILL	Fill a memory block with values
SEARCH	Search a memory block for a value
SETMEM	Change the values of

memory locations	
SETREG	Change the contents of a register
TEST	Examine memory area for invalid values

### Port I/O and Interrupt Commands

DIN	Display input port buffer values
DOUT	Display output port buffer values
INPORT	Set or alter input port status
INTERRUPT	Simulate an interrupt
NOINTERRUPT	Cancel pending interrupts
OUTPORT	Set or alter output port status
RIN	Rewind input file associated with input port
ROUT	Rewind output file associated with output port

### Symbol Commands

ADD	Create a symbol
DELETE	Delete a symbol from the symbol table
PRINTSYMBOLS	Display symbol, type, and address
SCOPE	Specify current module and procedure scope

### Utility Commands

CEXPRESSION	Calculate the value of an expression
ERROR	Set include-file error handling
HELP	Display on-line help screen
INCLUDE	Read in and process a command file
JOURNAL	Record a debugger session in a file
LOG	Record debugger commands and errors in a file
MODE	Select debugger mode (high or assembly)
OPTION	Set debugger options for this session
PAUSE	Pause simulation
RESET	Simulate microprocessor reset
RESTART	Restart program counter to the program starting address
STARTUP	Save the default start-up options

### Macro Commands

DEFINE	Create a macro
SHOW	Display the macro source

### Viewport Commands

VACTIVE	Activate a viewport
VCLEAR	Clear data from a viewport
VCLOSE	Remove a user-defined viewport or screen
VMACRO	Attach a macro to a viewport
VOPEN	Create a screen or viewport or change sizes
VSCREEN	Activate a screen
VSETC	Set the cursor position for a viewport
ZOOM	Increase or decrease the size of a viewport

### Function Key Commands

VACTIVE-1	Activate the next viewport (counter clockwise)
VACTIVE+1	Activate the next viewport (clockwise)
MODE	Change debugging mode (assembly/high)
ZOOM	Increase or decrease the size of a viewport
HELP	Access on-line help
VSCREEN	Change the active screen
BACK UP	Back up one command
STEP	Execute one machine instruction or source line
STEP OVER	Step, but execute through procedures

### In-Circuit Emulator Commands

ICE	Communicate with in-circuit emulator (ICE)
NOICE	Return to debugger command mode

### RESTRICTIONS

- Up to 4 instruction breakpoints are allowed simultaneously.
- When an instruction has a breakpoint set, control is returned to XRAY after this instruction has been executed. However, the information about the break address returned is correct, so user actions (like macros) are executed as expected.
- In the current release, the TRACE capability of the SDS 8051 is supported by means of the ICE/NOICE command. The SDS 8051 TRACE information can be recorded in a journal file, by means of the JOURNAL command. The next release of XRAY51 will support emulator tracing within the debugger.

# Symbolic debugging package XRAY51 for the SDS 8051 emulator

OM4129

## HARDWARE/SOFTWARE REQUIREMENTS

XRAY51 software is supplied on 5 $\frac{1}{4}$ " diskettes (3 $\frac{1}{2}$ " diskettes are available on request) together with the documentation: the *XRAY51 User's Guide*, *XRAY51 Installation Guide* and the *XRAY51 Reference Manual*. XRAY51 will run on an MS-DOS computer equipped with a hard disk and at least 512kbyte RAM.

To work with XRAY51, you will also need:

- An ASM51 Intel-compatible relocatable cross-assembler and LINK51 Intel-compatible linking loader (version 1.0 or later).

If you want to use the high-level language capabilities of XRAY51, the following PL/M-51 compiler is required:

- PLMTI51 (version 2.0 or later) with the ASM51 assembler

These products can be ordered from Philips Semiconductors. See Ordering Information.

Contact your local Philips representative for specific enquiries. Ask for the following leaflets:

*Stand-alone debug station for 80C51/8051-based systems*, ordering code 9398 366 00011;

*PL/M-51 compiler package for 80C51/8051-based systems*, ordering code 9398 366 20011;

*Cross-assembler package for 80C51/8051-based systems*, ordering code 9393 366 30011.

## SOFTWARE MAINTENANCE AND SUPPORT

After a 90-day warranty period, in which all support will be given free of charge, a software update and support agreement can be taken out with Philips for a modest annual fee. Philips realizes your need for fast and comprehensive support, and with a support license, you get direct access to a development team that can solve your problems.

## ORDERING INFORMATION

TYPE NUMBER	DESCRIPTION
OM4129	XRAY51 high-level language debugger for the SDS 8051; MS-DOS
<b>Related Products</b>	
OM4142	Cross-assembler package comprising the ASM51 assembler, and the LINK51 linker
OM4144	Compiler package comprising the PLMTI1

Orders can be placed via your local Philips Semiconductors sales representative. May be purchased from BSO Tasking, Tel. (800)458-8276 or (617)320-9400.

<sup>TM</sup> Intel is a trademark of Intel Corporation.  
MS-DOS is a trademark of Microsoft Corporation.

## 8051 C cross-compiler

OM4136

## INTRODUCTION

The BSO/Tasking C-51 cross-compiler (OM4136) offers a new approach to high-level language programming for the 8051 family. It is a very powerful combination of our extensive 8051 knowledge and our ANSI compliant C compilers. The result of this combination is a compiler that is fast, efficient and supports all the members of the 8051 family.

## FEATURES

- Supports:
  - Full ANSI Standard
  - All members of the 8051 family
  - Reentrant programming
  - Extremely efficient and powerful pointer arithmetic
  - Inline assembly programming
  - Inline expansion of predefined functions (`_lrol`, `_rrol`, etc.)
  - Full Intel OMF51 and IEEE695 version 4.0 object formats
  - Data overlay mechanism
  - Full calling interface (with parameter passing) to BSO/Tasking PL/M-51 (OM4144)
  - Fast 8-bit *char* arithmetics
  - C-level access to chips' SFRs
  - C-level bit-type and interrupt
- Generates Intel-compatible assembly source
- Outputs symbolic information for source level debugging using XRAY51
- Produces very efficient, fast executing and reliable code
- Code is ROMable
- Supports 4 memory models to best meet application requirements
- Separate compilation of program modules
- Embodies integral standard preprocessor
- Is one-pass, fast and compact (no intermediate code or files)
- 3-layer design simplifies maintenance
- Complete set of UNIX-like compiler options
- Comes with C library and run-time support, including I/O calls (+ *printf*), memory management, floating point, math and arithmetic functions
- C libraries in source code included for learning and tuning purposes
- The OM4136 product is for IBM PC DOS as host. From BSO/Tasking, the

C-compiler is also available on the hosts: (micro)VAX (VMS or Ultrix), IBM PC (Xenix), IBM PS/2 (OS/2, AIX), IBM RISC System/6000, SUN-3/4 (SUN-OS), HP9000/300 (HP-UX) and Apollo (UNIX)

- Generates reentrant and relocatable code and relocatable data
- Same calling sequence on all hosts
- Same source and generated assembly code can be used on all hosts
- Automatic installation and self test
- Benchmarks and application notes are available on request

## C LANGUAGE

Although intended as a general purpose high-level programming language, C is perhaps most powerful in the area of real-time system programming for embedded microcontrollers.

ANSI C features economy of expression, is well defined, has an improved type checking, supports structured and modular programming and has a rich set of data- and operator-types that map to virtually every single address space microcontroller/processor. Since the operators and type definitions generally match well with the instruction sets and word lengths of most microcontrollers, C is very efficient in terms of code size and execution speed.

## COMPILER TECHNOLOGY

The 8051 C compiler was built using the latest compiler technology, including optimization and multi-layering. It is essentially a one pass compiler, translating on a function-by-function basis, achieving very fast compilation and a large span for the optimization process. A significant effort was made to obtain a well-defined, layered product.

The top layer basically consists of a lexical and grammar analyzer with co-routine structures to serve both the pre-processing and compilation needs. The grammar analyzer is produced by an LALR(1) parser generator, eliminating a potential source of error.

A second layer separates the actual code generation layer from the top layer and is merely intended to simplify re-targeting of the compiler to other target processors. It was designed to represent the translated program independent from the language and the target. This is where most of the optimizations are done.

The code-generation layer is driven both by the second layer and by a set of tables and rules reflecting the target assembly language and the behavior of individual instructions.

## IMPLEMENTATION DEPENDENT DATA

This section outlines features of the C-51 compiler which, for the most part, are specific to the 8051 family of microcontrollers.

## Compiler Options

The following options are supported:

- a specify function parameter size
- b specify default register bank number
- C specify 8051 derivative type
- D define named identifier to the preprocessor
- E only run the preprocessor
- f read command line options from file
- g generate high-level debugging information for XRAY51 hl-debugger
- I force preprocessor to look for include files in the specified directory
- m specify memory sizes for static allocation checking
- M select memory model
- n write output to screen instead of output file
- o write output to named output file
- O controlled optimization
- r specify ROM size for optimal jump-type generation
- R control segment assignment
- S merge C source with generated assembly code
- s put strings in ROM only
- t do not produce module summary information
- U remove any initial definition of named identifier
- v do not generate code for interrupt vectors
- w suppress warning messages

## Data Sizes

Type	Size (in bytes)
bit	1 bit
char	1
short int	2
int	2
long int	4
float, (long) double	4 (IEEE single precision)
pointer type	1 or 2*

\*NOTE: Pointers to *data*, *idata* and *pdata* have a size of 1, whereas pointers to *rom*, *xdata* and functions have a size of 2.



## 8051 C cross-compiler

## 8051 C cross-compiler OM4136

## Code Optimization

The compiler intermediate layer performs general code-optimization and the code-generator does some final optimizations that are specific to the 8051.

## General Optimizations:

- Register allocation
- Branch optimization
- Dead code elimination
- Constant folding
- Arithmetic simplification

## Code Generator Optimizations:

- Store-copy optimization
- Peephole optimization

## Adaptation to Target Environment

Cross-compilers are used to develop embedded microcontroller applications, where the hardware environment in which they will run is not fixed in advance. The 8051 compiler allows flexible adaptation to different target environments:

- The compiler supports 4 different memory models. This allows the compiler to generate the best possible code for a particular hardware implementation.
- Interrupt service routines can be written in C or assembly (through special controls)

- Generated code is ROMable
- 8051 derivative is switch selectable
- User-controlled mapping of code and data

## Pragmas

The *#pragma* directive is intended to supply target dependant data to the compiler, without violating the C language. In C-51, pragmas are used to merge C lines with generated assembly, to control code generation for interrupt service routines and to support inline assembly programming.

## LIMITATIONS

The use of several C constructs is restricted to a maximum number, as proposed by ANSI. Both the preprocessor and compiler meet these 'environmental requirements of a conforming implementation'. As an additional constraint however, the number of statements in a function body is restricted to approximately 1800.

## DOCUMENTATION

The compiler is shipped with the 8051 C Compiler User Manual.

Additional information about the C language can be found in "The C Programming Language" (second edition), by B. Kerighan and D. Ritchie (1988, Prentice Hall).

## RELATED PRODUCTS

The C51 compiler (OM4136) is part of a complete programming and development package for the 8051 family of microcontrollers. This incorporates a HLL debugger, XRAY-51 (OM4129) in combination with Philips' SDS-8051 emulator, PL/M-51 compiler (OM4144), cross-assembler and (optimizing) linker (OM4142), all available from Philips Semiconductors.

## SOFTWARE MAINTENANCE AND SUPPORT

After a 90 days warranty period, in which all support will be given free of charge, a software support contract is available from BSO/Tasking at a modest annual fee. BSO/Tasking realizes the need for fast and thorough support in this complex area where a lot of time may be wasted before a (afterwards simple) problem is isolated and/or tackled. With a support license, you can use BSO/Tasking resources to solve your problem. You will get a direct access to experienced engineers. Furthermore, you receive (free of charge) the periodical *BSO/Tasking Newsletter*, regarding revisions, hints and documentation corrections.

## Memory Models

MODEL	DATA ALLOCATION	APPLICATION
<i>small</i>	static, in on-chip direct addressable RAM ( <i>data</i> )	fast programs in small environments
<i>auxpage</i>	static, in first page of external RAM ( <i>pdata</i> )	derivatives with 256 bytes on-chip "external" RAM
<i>large</i>	static, in external RAM ( <i>xdata</i> )	fast, non reentrant with large external RAM
<i>reentrant</i>	dynamic + static, in external RAM ( <i>xdata</i> )	large, reentrant programs in large environments

## ORDERING INFORMATION

TYPE NUMBER	DESCRIPTION
OM4136	8051 C-compiler package
<b>Related Products</b>	
OM4144	PLM-51 compiler
OM4142	8051 cross-assembler and linker
OM4129	XRAY51 high-level language debugger for the SDS 8051

Orders can be placed via your local Philips Semiconductors sales representative. May be purchased from BSO Tasking, Tel. (800)458-8276 or (617)320-9400.

Sourced from BSO/Tasking Software B.V., Amersfoort, The Netherlands, which holds all intellectual property rights for this software.

<sup>TM</sup> Intel is a trademark of Intel Corporation.  
MS-DOS and XENIX are trademarks of Microsoft Corporation.  
XRAY is a trademark of Microtec Research Inc.  
UNIX is a trademark of AT&T Bell Laboratories.  
VAX, microVAX, VMS and Ultrix are trademarks of Digital Equipment Corp.  
HP9000 and HP-UX are trademarks of Hewlett Packard Co.  
SUN-3 and SunOS are trademarks of Sun Microsystems Inc.  
IBM PC is a trademark of International Business Machines Corp.

## Cross-assembler package for 80C51/8051-based systems

## OM4142 (ASM51)

This package comprises the cross-assembler for translating 8051 assembly language programs into relocatable object code:

- The ASM51\* macroassembler which accepts Intel-compatible assembler source programs and produces relocatable ".OBJ" object files. An absolute or executable "A.OUT" load-image is obtained using the LINK51 linker.

The absolute object file can be modified to IEEE format to serve as input to the XRAY51 debugger.

### FEATURES

- Produces relocatable object code, listing files and diagnostic messages
- Accepts Intel-compatible source programs (ASM51)
- Supports most 8051-derived microcontrollers including the 80C51/31, 87C51, 8XC451/528/552/562/592/652/654/751/851, 8XCL410/710 and the 8051/31/52/32.
- Compatible with PL/M-51 compilers
- Conversion to IEEE object code format
- Compatible with XRAY51 High-Level/Assembly-Level Debugger
- Includes many utilities such as, Librarian, Cross-reference generator, object code-convertors
- Supports segment overlay at the Assembly level
- C and MPL (Macro Programming Language) compatible macro preprocessors included
- Separate linking phase.

### OPERATION

The input to both assemblers usually comes from a preprocessor which interprets preprocessor directives in the source program to deal with file inclusion, macro definition and replacement, conditional text inclusion, etc. Two preprocessors are available as separate programs, allowing the programmer to use either the C preprocessor directives or Intel's Macro Programming Language, or even a mixture of both.

The output may then be assembled using the ASM51. Depending on the selected member of the 8051 family, the ASM51 assembler enables or disables the names of special function registers that are applicable. For the ASM51, this is easily solved by target-dependent inclusion of a specific equate list in the source file.

The assembler translates a source program into relocatable object code using three different passes. The program syntax, assembler directives and user-defined symbols are checked and processed during the first pass. In the second pass, all generic forward jumps and calls are optimized. In the third pass, relocatable code is generated.

To obtain a single executable load image, all necessary relocatable objects, including library modules, are linked together using the proper linker (i.e., LINK51). This executable load image may then be converted to an ASCII file that may be downloaded into an EPROM programmer or an emulator.

The assembler is capable of generating symbolic debug information to accommodate information for the XRAY debugger.

### ASSEMBLER LIMITATIONS

The number of user-defined symbols and macro parameters is limited only by the available heap space. Save/restore nesting is restricted to 16 levels.

### DIFFERENCES BETWEEN

#### PHILIPS' ASM51\* AND INTEL'S ASM51

Unlike Intel's ASM51, which restricts the use of register equates to the A and R0-R7 registers, Philips' ASM51 puts no constraints on register name assignment. And it can optimize generic JMP and CALL instructions, even when they contain a forward reference. In such cases, Intel's ASM51 will always produce code for a LJMP and LCALL, which takes 50% more code.

Philips' ASM51 supports four new directives that are not available in the Intel ASM51:

- \$listall generate a listfile in every pass (not only in the final one). This improves diagnostics.
- \$(no)-optimize enable (default) or disable generic JMP/CALL optimization.
- \$debuginfo control symbolic debug information generation.

Since the \$gen, \$genonly, \$nogen, \$include and \$macro directives are already dealt with during the preprocessing stage, these can be ignored by Philips' ASM51.

Similarly, \$(no)-xref and \$(no)-symbols directives are ignored—report utilities are

available to accomplish the same task. And the \$workfiles is no longer useful and is ignored too.

Philips' ASM51 recognizes ?SYMB, ?LINE and ?FILE symbols that are used to pass debug information towards the object module. Philips' ASM51 recognizes the C-like #line directive to adjust the line number and file name. This directive is generated by both preprocessors to synchronize the output line with the original input. This improves the error diagnostics of Philips' ASM51 compared with those of its competitors, since error messages now refer to the proper (include) file and line number.

A powerful addition is the overlay() which gives the programmer full control of the section overlay strategy.

### UTILITY PACKAGE

Several utility programs are included for your convenience. Amongst these are a C preprocessor (cpp) and an Intel MPL-compatible preprocessor (mpl) to deal with the various preprocessor directives and macros.

In many cases, the format of the object code produced is not suitable for EPROM-programmers, emulators or debuggers. Therefore, several utilities are included to convert the code to Intel HEX (oct\_ihex), Motorola S0-S9 (oct\_srec) or IEEE-695 (oct\_ieee) or vice-versa (ocf\_ihex, ocf\_srec).

An optional utility package is available separately from Tasking Software B.V. It contains UNIX-like utilities to obtain a (cross-reference) list of all user-defined symbols in a program ('axref' and 'anm'), and the 'asize' utility to get information about the section sizes.

### HARDWARE/SOFTWARE REQUIREMENTS AND INSTALLATION

OM4142 software comes to you on 5 1/4" diskettes (3 1/2" diskettes are available on request) together with extensive documentation, including two Assembler Reference Guides, Preprocessor Manuals, Utility and Installation Guide. The software requires an MS-DOS computer with a hard disk and at least 512k byte RAM installed. Software installation is simple, well-documented and can be verified afterwards using an automatic verification program.

\* Sourced from BSO/Tasking Software B.V., Amersfoort, The Netherlands, which holds all intellectual property rights for this software.

## Cross-assembler package for 80C51/8051-based systems

## OM4142 (ASM51)

### RELATED PRODUCTS

The OM4142 assembler is part of a complete programming and development package for the 8051 family of microcontrollers. For both assemblers, there is an excellent PL/M-51 compiler available from Philips offering the convenience and benefits of high-level language programming, resulting in a dramatic increase of programmer productivity. Testing and debugging can be accelerated using the High-Level/Assembly-Level XRAY51 debugger. Contact your local Philips Semiconductors software sales office for more information. Ask for the following leaflets:

*Symbolic debugging package XRAY51 for the SDS 8051 emulator*, ordering code 9398 366 10011;

*PL/M-51 compiler package for 80C51/8051-based systems*, ordering code 9398 366 20011;

*Stand-alone debug station for 80C51/8051-based systems*, ordering code 9398 366 00011.

### SOFTWARE MAINTENANCE AND SUPPORT

After a 90-day warranty period, in which all support will be given free of charge, a software update and support agreement can

be taken out with Philips for a modest annual fee. Philips realizes your need for fast and comprehensive support, and with a support license, you get direct access to a development team that can solve your problems.

### ORDERING INFORMATION

TYPE NUMBER	DESCRIPTION
OM4142	8051 cross-assembler package comprising the ASM51 assembler, and the LINK51 linker
<b>Related Products</b>	
OM4144	PL/M-51 compiler package comprising the compiler
OM4129	XRAY51 high-level debugger for the SDS 8051

Orders can be placed via your local Philips Semiconductors sales representative. May be purchased from BSO Tasking, Tel. (800)458-8276 or (617)320-9400.

™ Intel is a trademark of Intel Corporation.  
XRAY is a trademark of Microtec Research Inc.  
MS-DOS is a trademark of Microsoft Corporation.  
UNIX is a trademark of AT&T Bell Laboratories..

## PL/M-51 compiler package for 80C51/8051-based systems

OM4144 (PLMTI51)

This package comprises the cross-compiler (PLMTI51\*) which efficiently translates programs written in the PL/M-51 language into 8051 Assembly. The PLMTI51 generates Intel-compatible source, which can be assembled and linked using Philips' ASM51 and LINK51 programs.

This compiler embodies the latest techniques. It is built upon a YACC-based parser. Intermediate code is built using tree structures that are optimized into new trees by the compiler before code-generation starts. An internal peephole optimizer further improves the density of the generated code.

It can be ordered under the Philips' type number OM4144.

### FEATURES

- State-of-the-art compiler: fast, single-pass, memory-based
- Fully compatible with the Intel PL/M-51 language definition
- Fully compatible with Philips' ASM51/LINK51 cross-assembler and linker
- Produces highly optimized code
- Supports most 8051-derived microcontrollers, including the 80C51/31, 87C51, 8XC451/528/552/562/592/652/654/751/851, 8XCL410/710/51 and the 8051/31/52/32
- Both compilers support the XRAY51 high-level language debugger, available from Philips
- Supports in-line assembly code
- Optional IEEE single-precision floating point package
- Includes several utility programs: pr, grep, aar, cpp, scf\_i51, ocf\_o51, oct\_ihex, oct\_ieee, oct\_srec, ocf\_ihex, ocf\_srec, makelib and pack
- Utility library included in source code
- Combiner performs optimization at load-file level (Philips' ASM51 only)
- Easy tailoring to application and target hardware
- Produces relocatable code and data
- Available for many hosts besides the IBM PC (MS-DOS), for example: (micro-)VAX, (VMS, Ultrix), HP9000/300 (HP-UX), SUN-3 (Sun-OS); available from BSO/Tasking Software B.V.
- Supports optimize(4): intermodule overlay optimization
- Supports 15 interrupts

- Automatic installation with self-test
- Benchmarks available on request

### PL/M-51

PL/M-51 is a structured, high-level programming language derived from the Intel PL/M-80 language and adapted to the specific capabilities of the 8051 family of single-chip microcontrollers. It supports Boolean processing and allows efficient access to all microcontroller hardware functions.

Software development in PL/M-51 combines the ease of programming in a high-level language with access to all of the 8051 I/O and memory—features that are normally only available to assembly language programmers.

The Philips' implementation of PL/M-51 is an extremely fast single-pass optimizing compiler that is fully compatible with the Intel PL/M-51 language definition.

### IMPLEMENTATION-DEPENDENT DATA

#### Data Types

- Data types BIT, BYTE and WORD are allowed for variables, arrays, structures or combinations thereof.
- Memory types MAIN, IDATA, REGISTER, AUXILIARY and CONSTANT are supported.
- Data can be stored at fixed locations using AT, or stored dynamically using BASED pointer variables.

#### Procedures

- BIT, BYTE or WORD typed procedures, returning a value upon completion
- Untyped procedures, invoked with a CALL statement
- Optional procedure attributes:
  - USING(n), specifies the register bank (n) to be used by the procedure.
  - INTERRUPT(n), defines an interrupt procedure for interrupt (n).

#### Statements

- Control statements: do while ... end, do ... to ... by ... end
- Conditional statements: if ... then ... else ...
- Miscellaneous statements: do ... end, do case ... end, call, goto, enable/disable

### Library Routines

#### Internal (Built-in) Procedures:

LENGTH	PROPAGATE
WORD	ROL
EXPAND	SCR
SHR	SIZE
SCL	BOOLEAN
TIME	SHL
LAST	ROR
DOUBLE	TESTCLEAR

#### PLM51.LIB:

Library routines used by the compiler.

#### UTIL51.LIB or util51.oa, util51:

Assembler utility libraries for both compiler versions, consisting of procedures for string manipulation. It contains the routines MOV, RMV, CMP, FNDB, FNDW, SKPB, SKPW, SETB and SETW for each memory type and for each register bank.

### Re-entrancy

The generated code is not re-entrant, because PL/M-51 is not defined as a language with re-entrant procedures. Because of the limited size of internal RAM, the local and formal parameters of procedures are not put on the stack, but on static areas, which can be overlaid by the compiler using \$OPTIMIZE(3).

### Compiler Controls

ROM(s), REGISTERBANK(n), OPTIMIZE(n), (NO)INTVECTOR, (END)ASM, INCLUDE, SAVE/RESTORE, (NO)LIST, (NO)SOURCE, (NO)CODE, (NO)DEBUG, (NO)OBJECT, SET/RESET, EJECT and IF/ELSIF/ELSE/ENDIF

### Differences Between Philips' and Intel's PL/M-51

- Philips' PL/M-51 allows bit-structures to be ATed at byte-variables in bit-addressable memory
- Philips' PL/M-51 supports floating point
- Invocation and compiler controls are mostly different
- Object modules are not generated
- Error messages are mostly different
- Different assembly language interfacing (only for Philips' plm51)
- Compiler limits always the same or better than Intel's

\* Sourced from BSO/Tasking Software B.V., Amersfoort, The Netherlands, which holds all intellectual property rights for this software.



## PL/M-51 compiler package for 80C51/8051-based systems

OM4144 (PLMTI51)

### Code Optimization

Five levels of optimization are supported:

- Level 0: performs folding of constant expressions and of address calculations, in case a constant offset exists.
- Level 1: performs all of level 0 plus elimination of unreachable code, strength reduction of expressions and partial condition evaluation at runtime.
- Level 2: performs all of levels 0 and 1 plus machine code (peephole) optimizations and register history.
- Level 3: performs all of levels 0, 1, and 2 plus automatic overlaying of on-chip RAM variables.
- Level 4: performs all of the other levels plus support of intermodule overlay of on-chip data.

### General Optimizations

- Store-copy optimization
- Local constant propagation
- Register allocation
- Peephole optimization
- Dead code elimination
- Constant folding
- Index simplification

### Object-code Optimizations

- Branch optimization (sjmp, ajmp, ljmp, acall, lcall)
- Effective address optimization

### 8051-Specific Optimizations

- Optimal use of the range of address modes of the 8051 family
- Overlay of local data and formal parameters done by \$OPTIMIZE(3)/(4)

### Easy Adaptation to Target Environment

Cross-compilers are used to develop embedded microprocessor applications, where the hardware environment in which they will run is not fixed beforehand. Adaptation to the target hardware environment takes place via the files headx. These files are used to define all system-dependent set-ups such as the

power-on-restart vector, the initial stackpointer value, the definitions of segments needed by the PL/M-51 code and the mapping of those segments to the physical addresses. The files can be modified by the user to match his specific needs.

The PL/M-51 cross-compiler can accommodate different target environments, because:

- The location of the stack is held in a special target set-up file (head\_xx file) which can be changed to match your requirements
- This target set-up file also allows general housekeeping tasks to be executed before program start-up
- Simple interfacing to target operating systems and low-level I/O and system routines
- Code and data segments can be placed anywhere in the 64kbyte data and address space of the 8051 processor
- Efficient calls to library routines can be in PL/M or assembler (in-line)
- A user-written interrupt handler can be made by specifying \$NOINT/VECTOR
- The generated code is ROMable.

### Restrictions

The PL/M-51 compiler has a few restrictions listed here for completeness:

- Nesting of all LITERALLY invocations: 8
- Nesting of INCLUDE controls: 8
- Nesting of blocks: 32
- Number of elements in a factored list: no limit
- Number of characters in an input line: 160
- Number of switch names (conditional compilation): 20
- Length of a string constant: 254
- Number of cases in a DO CASE block: 84
- Number of EXTERNAL items: no limit
- Number of non-EXTERNAL procedures in module: no limit
- Number of names in a module: memory dependent

### UTILITY PACKAGE

Several utility programs are included for your convenience. Amongst these are a C preprocessor (cpp), an Intel MPL compatible preprocessor (mpl)—both part of the 8051 Cross-Assembler—to deal with the various preprocessor directives and macros. Pagination and pattern search commands 'pr' and 'grep' come with the PL/M-51 compiler package.

In many cases, the format of the produced object code is not suitable for EPROM-programmers, emulators or debuggers. Therefore, several utilities are included (in the 8051 cross-assembler) to convert the code to Intel HEX (oct\_ihex), Motorola S0-S9 (oct\_srec) or IEEE-695 (oct\_ieee) or vice-versa (ocf\_ihex, ocf\_srec) or to archive object modules (aar).

### OPTIONAL SOFTWARE

An optional utility package is available separately from BSO/Tasking Software B.V. It contains UNIX-like utilities to obtain a (cross-reference) list of all user-defined symbols in a program ('axref' and 'anm'). Part of this package is the 'asize' utility to get information about the section sizes, 'asort' to sort or merge files, 'astrip' to remove symbols and relocation information, 'adump' to display the contents of an object file.

Also available is an IEEE floating-point library for the 8051 family.

### HARDWARE/SOFTWARE REQUIREMENTS AND INSTALLATION

OM4144 software comes to you on 5 1/4" diskettes (3 1/2" diskettes are available on request) together with the extensive *PL/M-51 Application Manual*.

The software requires an MS-DOS (Rel. 3.0 or higher) computer with a hard disk and at least 512kbyte RAM installed. Software installation is simple, well-documented and can be verified afterwards using an automatic verification program.



## PL/M-51 compiler package for 80C51/8051-based systems

OM4144 (PLMTI51)

### RELATED PRODUCTS

The OM4144 compilers are part of a complete programming and development package for the 8051 family of microcontrollers, a package which includes a cross-assembler and (optimizing) linker. Testing and debugging can be accelerated using the high-level/assembly-level XRAY-51 debugger in combination with Philips' SDS 8051 emulator, or the 8051 Simulator (available from Tasking software B.V.). Contact your local Philips Semiconductors sales office for more information.

Ask for the following leaflets:

*Symbolic debugging package XRAY51 for the SDS 8051 emulator*, ordering code 9398 366 10011;

*Cross-assembler package for 80C51/8051-based systems*, ordering code 9398 366 30011;

*Stand-alone debug station for 80C51/8051-based systems*, ordering code 9398 366 00011.

More information about the PL/M-51 Language can be found in the *PL/M-51 User's Guide*, available from Intel (ordering code: 121966-003).

### SOFTWARE MAINTENANCE AND SUPPORT

After a 90-day warranty period, in which all support will be given free of charge, a

software update and support agreement can be taken out with Philips for a modest annual fee. Philips realizes your need for fast and comprehensive support, and with a support license, you get direct access to a development team that can solve your problems.

### ORDERING INFORMATION

TYPE NUMBER	DESCRIPTION
OM4144	PL/M-51 compiler package comprising the PLMTI1 compiler
<b>Related Products</b>	
OM4142	8051 cross-assembler package comprising the ASM51 assembler, and LINK51 linker
OM4129	XRAY51 high-level language debugger for the SDS 8051

Orders can be placed via your local Philips Semiconductors sales representative. May be purchased from BSO Tasking, Tel. (800)458-8276 or (617)320-9400.

<sup>TM</sup> Intel is a trademark of Intel Corporation.  
MS-DOS and XENIX are trademarks of Microsoft Corporation.  
XRAY is a trademark of Microtec Research Inc.  
UNIX is a trademark of AT&T Bell Laboratories.  
VAX, microVAX, VMS and Ultrix are trademarks of Digital Equipment Corp.  
HP9000 and HP-UX are trademarks of Hewlett Packard Co.  
SUN-3 and SunOS are trademarks of Sun Microsystems Inc.  
IBM PC is a trademark of International Business Machines Corp.

## Stand-alone debug station for 80C51/8051-based systems

# SDS 8051

### THREE CONFIGURATIONS TO SUIT YOUR 8051 PROJECTS

For efficient, accurate software and hardware development and integration, there is no substitute for fully transparent, real-time emulation. But development must be cost-effective, and the key to that is to have one emulation unit which can fit exactly into all your development projects irrespective of their scope and stage of development.

If you are developing with the 8051 family, such a unit is the SDS 8051.\* Three configurations suit all your projects without requiring modifications or extras:

#### Stand-Alone Operation

A VDU Terminal is all that is needed to integrate and debug with the SDS 8051. And because you can do basic emulation without putting demands on computer resources, the smallest user as well as the big development team can take advantage of the SDS 8051.

#### Connection to a Host Computer

Most development tasks will, however, need the power of a host computer or Microcomputer Development System, and Philips' SDS (Stand-alone Debugging Station) can work with both. The process of writing the software and designing the hardware can be done on the computer, with an SDS used to debug and integrate the software with the hardware. SDS 8051 supports a wide range of hosts. It has its own in-line assembler in firmware, but you may also use the separate MS-DOS cross-assembler.

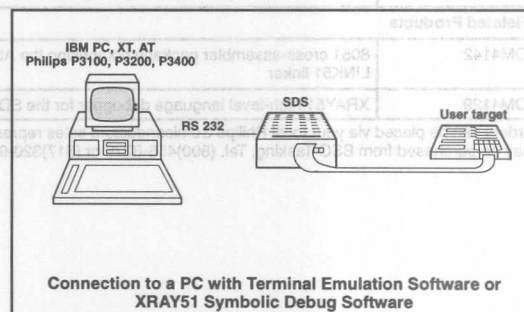
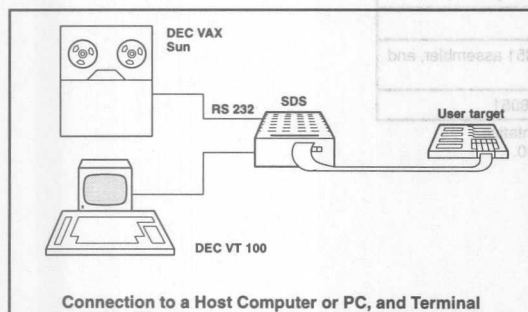
#### Connection to a PC

The SDS 8051 operates with an IBM PC or XT, or compatible, e.g., Philips P3100 series, so you can use it in your usual work environment. The SDS can be operated from a PC using readily available terminal emulation software, or the XRAY<sup>TM</sup>51 symbolic debugging package, the latter allowing you to use your own source labels on screen in SDS displays and commands, and having a DOS toggle switch.

If you work in a development team, shared access to files and programs is of course essential to maximize productivity. Programs can be downloaded from any host computer to the SDS memory, using a PC or terminal as a workstation to operate the SDS.

### FEATURES

- Real-time, fully transparent emulation
- Works stand-alone from a VDU terminal, PC or other computer
- Full hardware emulation using a dedicated probe—operates according to the exact specifications of the target microcontroller, including maximum speed
- Interfacing signals to external equipment
- Full assembly-level debugging and HLL debugging
- Single-step and breakpoint facilities
- Large trace memory with hardware qualifiers



\*SDS 8051: Generic name for Philips Stand-alone Debugging Station for the 8051 family of microcontrollers; for type numbers, see Ordering Information. The 8051 family includes the 80C51/31, 87C51, 80CL51, 8XC053/054/451/528/550/552/562/592/652/654/751/752/851, the 8XCL410/710 and the 8051/31/52/32. Debugging stations are also available for Philips 8400 microcontroller family and the 5010/5011 digital signal processors.

## Stand-alone debug station for 80C51/8051-based systems

SDS 8051

### UNEQUALLED EMULATION

SDS 8051 provides complete, high-quality emulation for development with the 8051 family of microcontrollers. Standard features include:

#### Fully Transparent Real-Time Emulation

You can develop on the SDS 8051 with the chip running in real time—no stretched clock cycles or wait states to disrupt system timing in the prototype. And emulation is fully transparent—since no resources from the 8051 memory, I/O or interrupt space are used for monitoring emulation, all are available to the target and user program.

#### In-Circuit Emulation or Simulation

The SDS 8051 is versatile—you can tailor operation to suit the state of the prototype. If no prototype hardware is available, the SDS may be used for simulation, so the software can still be tested. The emulation mode runs the program in the prototype so far as it has been developed, with resources transferred in stages to the prototype. Hardware, software and their integration are fully tested.

#### Full Real-Time Hardware Emulation and Breakpoint Setting

The SDS 8051 has an 80C51 (or derivative) bond-out chip in the emulation probe. This, and only this, can ensure that hardware emulation is truly real-time, and it also allows you to set hardware breakpoints. These can be set on any combination of addresses, register values or branch instructions,

allowing you to investigate program flow in detail and to debug very quickly. When a breakpoint condition is met, the entire CPU is frozen and, besides the full interrupt status, the status of all timers (frozen by a special bond-out chip feature) is displayed.

#### Alterable Memory and Processor Registers

For really quick fault-finding, the SDS allows you to alter, and to display, memory and CPU register values as required, allowing parts of the program to be repeatedly tested using convenient values.

In addition, the SDS has an emulator-resident in-line assembler which is particularly useful when changing your program—there being no need for repeated up-loading and down-loading.

#### Disassembly

Of course, there is no need to remember binary code references when developing software with the SDS 8051—instructions are entered in assembly language. In addition, on-board memory can be disassembled into the originally programmed instruction mnemonics.

#### Trace Memory

SDS 8051 has a 2048-line trace memory for quick checking of the program flow. The display shows the address, opcode/operand, disassembly, instruction status (such as RD OPC, WR) and the contents of microcontroller ports and/or eight user test clips.

### Hardware Qualifier Bits

The trace memory has qualifier bits for selecting the information to be captured in the trace memory. These bits enable cycles to be selected, for example, capture only on fetches from emulation memory, or on interrupt acknowledge cycles. Selecting the information before capture overcomes the drawback of software qualifiers which select the information afterwards, relying on luck for it to be in the trace memory!

### Debugger Commands

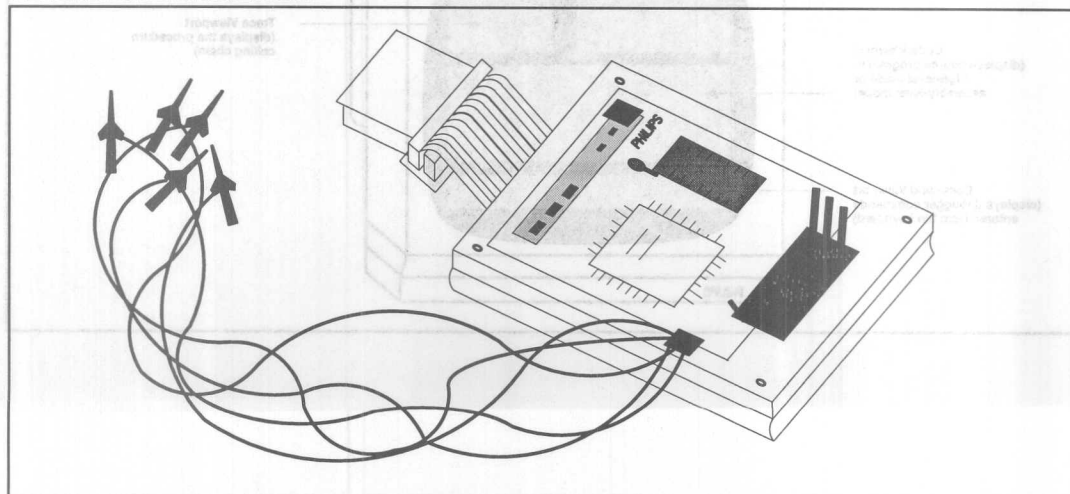
The microprocessor development command language employed by the SDS 8051 has been designed for ease of use. Commands form a subset of, and are similar to, the widely used ICE™ language.

When the XRAY51 symbolic debugging package is used, the commands are the XRAY commands. These and the SDS commands are listed below (at the end of this description).

### Software Packages

The XRAY51 high-level debugger, and cross-assemblers and cross-compilers for the SDS 8051 are described on the next page. Each package supports most 8051-derived microcontrollers including the 80C51, 83C053, 83C054, 80C451, 80C528, 80C550, 80C552, 80C562, 80C592, 80C652, 80C654, 83C751, 83C752, 80C851, 80C852, and 8XCL410/710.

### TYPICAL EMULATION PROBE



## Stand-alone debug station for 80C51/8051-based systems

# SDS 8051

### SYMBOLIC DEBUGGING PACKAGE (XRAY51)

The XRAY51 debugger helps to locate programming errors in the source code of C, PL/M or assembly language programs for the 8051 family. With XRAY51, the user can isolate these errors by controlling and monitoring program execution using the same high-level or assembly level terms, definitions and structures found in the original source program. For example, the user can single-step through the program a specified number of microcontroller instructions or high-level language lines. Variables can be accessed with respect to the source language in which they had been defined. Operating XRAY51 in Assembly mode, the user can manipulate the contents of all processor registers. Only those registers that are part of the selected 8051 derivative are allowed to be accessed.

Macros may be defined that can execute complex user command procedures and provide a variety of complex breakpoints.

When debugging with XRAY51, the user can examine the contents and modify the value of

any variable, compute the value of C, PL/M source language expressions and assembly level address expressions, and define, remove, or display symbols.

Command files can be used to direct XRAY51 to read or write simulated microprocessor input/output from or to a file, allowing easy implementation of automated test sequences. Command files enable scripts of debugger commands to be processed automatically without the need of user interaction.

XRAY51 also allows a virtually unlimited number of user-defined windows.

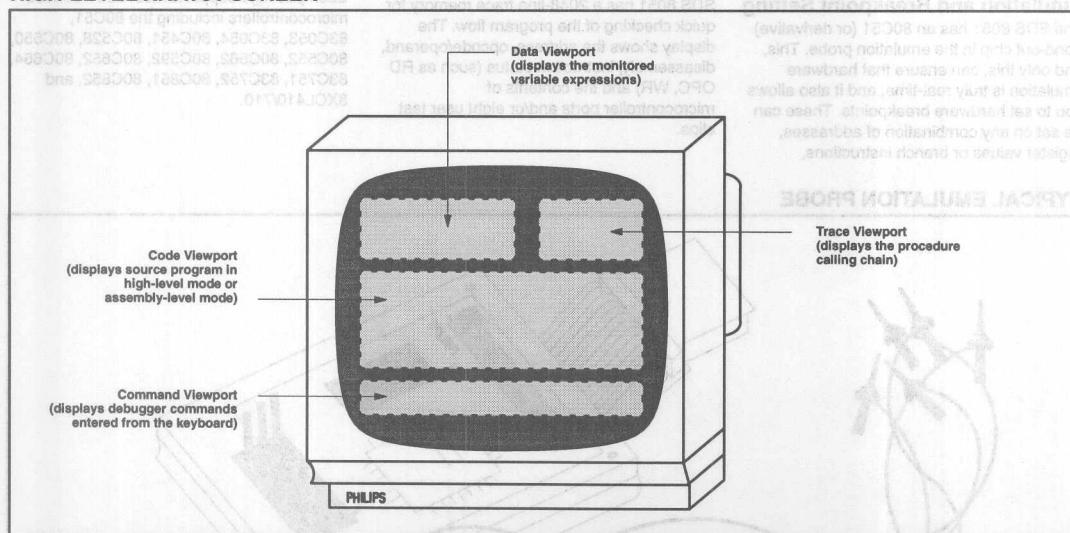
A simulator version of XRAY51 having the same features as the emulator is available.

### Features

- Integrated C, PL/M-51 source-language and assembly-language debugging; toggle by function key at any time
- Window-oriented display with well-organized segregation of debugging information

- Symbolic debugging with C, PL/M-51 variables and C, PL/M-51 statements
- Simple and complex breakpoints
- Single-step execution
- User-definable screens and viewports—ability to write selected information
- Command macros
- Breakpoint macros (limited to the number of breakpoints of the SDS 8051)
- Command and breakpoint macros may contain C statements, including: FOR, WHILE DO, PRINTF, and FPRINTF
- On-line context-sensitive help
- Command files
- Output logging
- Fully supports SDS 8051 with the latest firmware version
- High-level trace
- Transparent ICE mode for direct control of SDS 8051

### HIGH-LEVEL XRAY51 SCREEN



## Stand-alone debug station for 80C51/8051-based systems

SDS 8051

### CROSS-ASSEMBLER

The ASM51 cross-assembler\* is available for translating 8051 assembly language programs into relocatable object code:

It accepts Intel-compatible assembler source programs and produces relocatable '...OBJ' object files. An absolute or executable 'A.OUT' load-image is obtained using the LINK51 linker.

The assembler is a powerful development tool with many directives, so a source file is easily composed, and translation to code is fast.

The assembler directives support features such as the inclusion of ASCII strings, checking module length, forcing a start address, allocating bytes to labels used in current or other modules and module definition.

A very powerful feature is the macroprocessor which includes tools such as macro-substitution, file inclusion and conditional assembly.

### ASM51 Features

- Produce relocatable object code, listing files and diagnostic messages
- Accept Intel-compatible source programs
- Compatible with Tasking's PL/M-51 compilers
- Conversion to IEEE object code format
- Compatible with XRAY51 High-Level/Assembly-Level Debugger
- Include many utilities such as, Librarian, Cross-reference generator, object code-convertors
- Support segment overlay at the Assembly level
- C and MPL (Macro Programming Language) compatible macro preprocessors included
- Separate linking phase

### CROSS-COMPILERS

Two cross-compilers\* are available for efficiently translating programs written in the PL/M-51 language into 8051 Assembly:

- **PLM51:** generates Intel-compatible source, which can be assembled and linked using the ASM51 and LINK51 programs.
- **C-51:** generates Intel-compatible assembly source, which can be assembled and linked using ASM51 and LINK51.

Both compilers are always supplied together, allowing you to choose the one that suits your requirements best.

### PLM51 Features

- Fast, single-pass, memory-based compilers that are fully compatible with the Intel PL/M-51 language definition
- Fully compatible with ASM51/LINK51
- Produce highly optimized code
- Support the XRAY51 high-level debugger
- Support in-line assembly code
- Optional IEEE single-precision floating point package
- Include several utility programs: pr, grep, aar, cpp, scf\_i51, ocf\_o51, ocf\_ihex, ocf\_ieee, ocf\_srec, ocf\_ihex, ocf\_srec, makelib and pack
- Utility library included in source code
- Combiner performs optimization at load-file level
- Easy tailoring to application and target hardware
- Produce relocatable code and data
- Support optimize(4): intermodule overlay optimization
- Support 15 interrupts
- Automatic installation with self-test
- Available for many hosts besides the IBM PC (MS-DOS), for example: (micro-)VAX, (VMS, Ultrix), HP9000/300 (HP-UX), SUN-3 (Sun-OS); available from BSO/Tasking Software B.V.

### C-51 Features

- Supports ANSI C with powerful 8051 extensions:
  - *bit*-type to use the on-chip bit addressable area
  - *interrupt <number>* and *using <registerbank>* for interrupt servicing at C-level
- Generates Intel-compatible Assembly source

- Supports HLL-debugger XRAY51 (in combination with ASM51)
- Supports all members of the 8051 family
- Produces very efficient and reliable code
- Code is ROMable
- Easy migration from BSO/Tasking PLM-51 to BSO/Tasking C-51 through mixed language programming
- Supports 4 memory models to best meet application requirements
- Separate compilation of program modules
- Embodies integral standard preprocessor
- Is one-pass, fast and compact (no intermediate code or files)
- 3-layer design simplifies maintenance
- Complete set of UNIX-like compiler options
- Comes with C library and run-time support, including I/O calls (+ *printf*), memory management, floating point math, and arithmetic functions
- Available on many hosts: (micro)VAX (VMS or Ultrix), IBM PC (DOS, Xenix), IBM 6150, SUN-3/-4/-386i (SUN-OS), HP9000-300 (HP-UX), Apollo (UNIX)
- Generates re-entrant + relocatable code and relocatable data
- Same calling sequence on all hosts
- Same source and generated assembly code can be used on all hosts
- Automatic installation and self test
- Benchmarks and application notes are available on request.

\* Sourced from BSO/Tasking Software B.V. Amersfoort, The Netherlands



## Stand-alone debug station for 80C51/8051-based systems

SDS 8051

### XRAY51 AND SDS COMMANDS

#### XRAY51 Commands

XRAY51 uses a powerful command language that employs C language expressions. Note, all commands can be issued in an abbreviated form.

#### Session Control Commands

**HOST** Enter the host operating system environment  
**LOAD** Load an object module for debugging  
**QUIT** Terminate a debugging session

#### Execution and Breakpoint Commands

**BREAK** Set an instruction breakpoint  
**INSTRUCTION** Set an instruction breakpoint  
**CLEAR** Clear a breakpoint  
**GO** Start or continue program execution  
**GOSTEP** Execute macro after each instruction step  
**STEP** Execute a specified number of instruction lines  
**STEPOVER** Step, but execute through procedures

#### Display Commands

**DISASSEMBLE** Display disassembled memory (assembly mode)  
**DUMP** Display memory contents  
**EXPAND** Display all local variables of a procedure  
**FIND** Search for a string  
**FOPEN** Open a file or device for writing  
**FPRINTF** Print formatted output to a viewport or file  
**LIST** Display source code  
**MONITOR** Monitor variables  
**NEXT** Find next occurrence of a string  
**NOMONITOR** Discontinue monitoring variables  
**PRINTF** Print formatted output to a command viewport  
**PRINTVALUE** Print the value of a variable

#### Memory Commands

**COMPARE** Compare two blocks of memory  
**COPY** Copy a memory block  
**FILL** Fill a memory block with values  
**SEARCH** Search a memory block for a value  
**SETMEM** Change the values of memory locations  
**SETREG** Change the contents of a register  
**TEST** Examine memory area for invalid values

#### Port I/O and Interrupt Commands

**DIN** Display input port buffer values  
**DOUT** Display output port buffer values

**IMPORT** Set or alter input port status  
**INTERRUPT** Simulate an interrupt  
**NOINTERRUPT** Cancel pending interrupts  
**OUTPORT** Set or alter output port status  
**RIN** Rewind input file associated with input port  
**ROUT** Rewind output file associated with output port

#### Symbol Commands

**ADD** Create a symbol  
**DELETE** Delete a symbol from the symbol table  
**PRINTSYMBOLS** Display symbol, type, and address  
**SCOPE** Specify current module and procedure scope

#### Utility Commands

**CXPRESSION** Calculate the value of an expression  
**ERROR** Set include-file error handling  
**HELP** Display on-line help screen  
**INCLUDE** Read in and process a command file  
**JOURNAL** Record a debugger session in a file  
**LOG** Record debugger commands and errors in a file  
**MODE** Select debugger mode (high or assembly)  
**OPTION** Set debugger options for this session  
**PAUSE** Pause simulation  
**RESET** Simulate microprocessor reset  
**RESTART** Restart program counter to the program starting address  
**STARTUP** Save the default startup options

#### Macro Commands

**DEFINE** Create a macro  
**SHOW** Display the macro source

#### Viewport Commands

**VACTIVE** Activate a viewport  
**VCLEAR** Clear data from a viewport  
**VCLOSE** Remove a user-defined viewport or screen  
**VMACRO** Attach a macro to a viewport  
**VOPEN** Create a screen or viewport or change sizes  
**VSCREEN** Activate a screen  
**VSETC** Set the cursor position for a viewport  
**ZOOM** Increase or decrease the size of a viewport

#### Function Key Commands

**VACTIVE-1** Activate the next viewport (counter clockwise)  
**VACTIVE+1** Activate the next viewport (clockwise)  
**MODE** Change debugging mode (assembly/high)

**ZOOM** Increase or decrease the size of a viewport  
**HELP** Access on-line help  
**VSCREEN** Change the active screen  
**BACK UP** Back up one command  
**STEP** Execute one machine instruction or source line  
**STEP OVER** Step, but execute through procedures

#### In-Circuit Emulator Commands

**ICE** Communicate with in-circuit emulator (ICE)  
**NOICE** Return to debugger command mode

### SDS Commands

#### Power-Up Commands

**RESET** Back to initialization

#### Program Execution Commands

**BRn** Break at given address (n = 0, 1, 2, 3)  
**BRR** Break within given address range  
**BRB** Break at branch instruction  
**BV** Break on internal RAM value  
**GO** Initiates program execution (specified addresses)  
**STEP (FROM)** Executes single instruction or instructions (from a specified address)  
**UD** User-defined memory address on display  
**TRACE** Display executed instruction flow (real time)  
**INT** Display interrupt enable, priority and status of all interrupt sources

#### Memory Access Commands

**DBYTE** Displays specified byte from internal data memory  
**XBYTE** Displays specified byte from external data memory  
**CBYTE** Displays specified byte from code memory  
**RBYTE** Displays specified byte from on-chip register memory  
**RBIT** Displays specified bit from on-chip bit-addressable memory  
**ASM** Assemble single instruction mnemonic into program memory  
**DASM** Disassemble memory values into mnemonics

#### Memory Set Commands

All registers can be set and displayed by commands equal to their names (PSW, SCON, P3, TH1, TL1, TL0, etc.)

#### Serial I/O Interface Commands

**SAVE** Copies data from SDS 8051 program memory to host disk file  
**LOAD** Transfers file from host to SDS 8051

# Stand-alone debug station for 80C51/8051-based systems

SDS 8051

**SDS 8051 SPECIFICATION**

RS232C	<ul style="list-style-type: none"> <li>Two ports</li> <li>Baud rate selectable from 300 to 19200 baud</li> <li>Download file format: Intel HEX</li> <li>Recognize Xon/Xoff</li> </ul>
Trace memory	<ul style="list-style-type: none"> <li>2048 lines deep, 64 bits wide</li> <li>Internal/external code memory fetches; data from all ports, lables, user test clips</li> <li>Selective tracing on cycle type</li> </ul>
Emulation memory	<ul style="list-style-type: none"> <li>64 kbytes</li> <li>No wait states</li> </ul>
Clock speed	<ul style="list-style-type: none"> <li>Up to 20MHz, real time</li> </ul>
Power down	<ul style="list-style-type: none"> <li>Supports power-down and idle mode</li> </ul>
Signals to external equipment	<ul style="list-style-type: none"> <li>Output from SDS: <ul style="list-style-type: none"> <li>ALE: indicates valid address</li> <li>CLK: indicates opcode read</li> <li>PSENE: indicates byte read</li> <li>EMUL: indicates running user program</li> </ul> </li> <li>Input to SDS: <ul style="list-style-type: none"> <li>EXTBRK: stop emulation by external pulse</li> </ul> </li> </ul>
Size	<ul style="list-style-type: none"> <li>300 × 66 × 235 mm (W × H × D)</li> </ul>
Weight	<ul style="list-style-type: none"> <li>5 kg (approx.)</li> </ul>
Power supply	<ul style="list-style-type: none"> <li>110/220V AC, 50/60 Hz</li> </ul>
Cables	<ul style="list-style-type: none"> <li>Mains cable; RS232C/V24 cable for connection to an IBM PC/XT</li> </ul>

**ORDERING INFORMATION<sup>1</sup>**

All of the 8051 development tools listed below are available from your local Philips Semiconductors sales office. The symbolic debugging package, cross-assembler and compiler are sourced from Tasking Software B.V., Amersfoort, The Netherlands, which holds all intellectual property rights for these products.

**Stand-alone debug station** OM4120S  
(excluding probe) for the 8051 family

**Symbolic debugging package XRAY51** OM4129  
**SDS-Window debugger** OM1089

**Cross-assembler for MS-DOS** OM4142  
(Tasking)

**Cross-assembler for MS-DOS** OM1024  
(2500)

**PL/M-51 compiler for MS-DOS** OM4144

**C-51 compiler** OM4136

**Emulation probes<sup>4</sup>:**  
for 8032/8052/80C32/80C52<sup>3</sup>: OM4111+  
(DIL/PLCC interface) OM4110

for 80C51/80C31/8051/8031: OM1092+  
(DIL/PLCC interface) OM1097

for 80CL51: OM1079  
(DIL interface)

for 8XC053/054: OM5054  
(SDIP interface)

for 8XCL410/411: OM1079  
(DIL interface)

for 8XCL781/782/52: OM5012+  
(DIL interface) OM1079

for 8XCL580: OM5004+  
OM1079

for 83CL168/167/268/267: OM4840+  
(SDK 64 interface) OM1079

for 8XC451: OM4123  
(PLCC interface)

for 8XC528/524: OM4111+  
(DIL/PLCC interface) OM4110

for 8XC550: OM5055+  
(DIL/PLCC interface) OM4110

for 8XC552/562: OM1092+  
(PLCC interface) OM1095

for 8XCE558 OM4271+  
(No QFP-80 adapter) OM4110

for 8XC592: OM4112+  
(PLCC interface) OM4110

for 8XCE598 OM4114+  
(No QFP-80 adapter) OM4110

for 8XC652/654: OM1092+  
(DIL/PLCC interface) OM1096

for 8XC751: OM1094  
(DIL/PLCC interface)

for 8XC752: OM5072  
(DIL interface)

for 8XC851: OM1092  
(DIL/PLCC interface)

for 83C852: OM4119

**Adapters:<sup>2</sup>**  
for 8XC451: 68-pin PLCC probe to 64-pin DIL socket OM4124

for 80C51: 40-pin DIL probe to 44-pin PLCC socket OM4125

for 80C51 DIL 40 types: DIL-40 socket to QFP-44 footprint: OM4117

for 8XC552/562 PLCC-64 socket to QFP-80 footprint: OM4116

for 8XCE598/558 QFP-80 footprint: OM4115

for 83C852: contactless card reader adapter OM4118/1

ISO contact card reader adapter OM4118/2

AFNOR contact and reader adapter OM4118/3

**Conversion kits:**  
for converting the OM1092 to an:  
8XC552/562 probe OM1095  
8XC652/654 probe OM1096  
80C31/80C51 probe OM1097

**NOTES:**

1. A minimum SDS configuration must include an OM4120S and an emulation probe. A minimum PC configuration must include 256 kbytes system memory running MS-DOS 3.0 (or later releases), one floppy disk drive and a monochrome monitor. However, we recommend using an IBM PC/XT (or compatible) with 640 kbytes RAM, hard disk drive, floppy disk drive and a color monitor. The SDS software can be supplied on 3<sup>1</sup>/<sub>2</sub>" or 5<sup>1</sup>/<sub>4</sub>" diskettes. Conversion kits require the OM1092.
2. Support for QFPs will be available in the near future.
3. Minor restrictions.
4. All probes 16MHz, except OM4123 (12MHz) and OM4119 (6MHz)

<sup>TM</sup> ICE is a trademark of Intel Corporation.  
XRAY is a trademark of Microtec Research Inc.  
MS-DOS is a trademark of Microsoft Corporation.  
VAX is a trademark of Digital Equipment Corporation.



# Section 9

## Package Outlines

### 80C51-Based 8-Bit Microcontrollers

#### CONTENTS

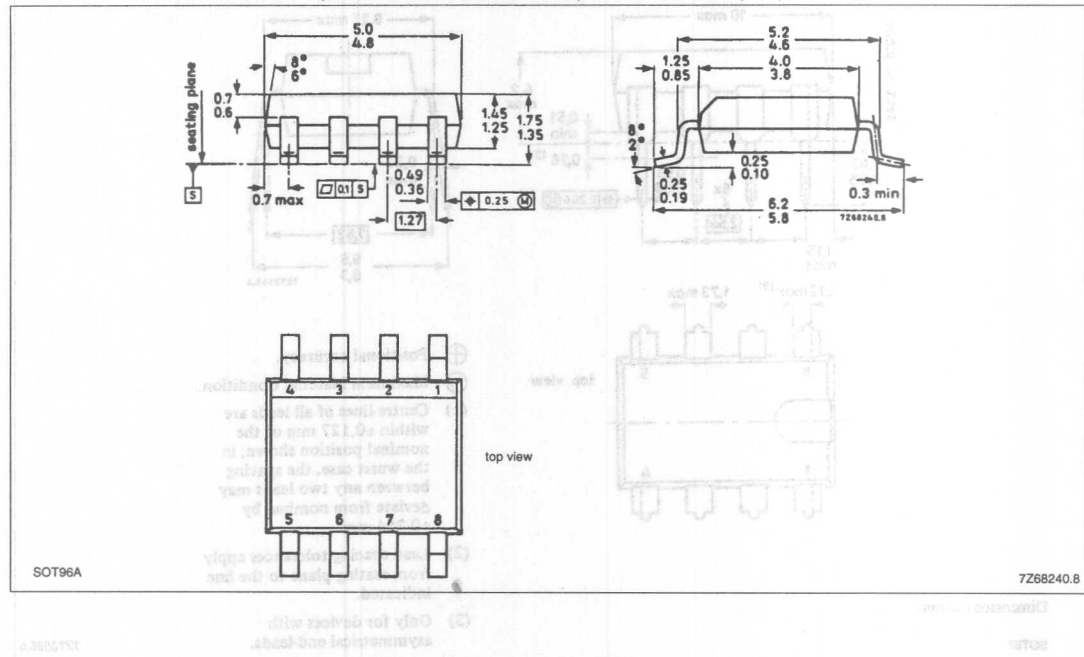
SOT96A	8-Pin Plastic SOL (Small Outline Large) Dual In-Line (D/T) Package . . .	1355
SOT97	8-Pin Plastic Dual In-Line (N/P) Package . . . . .	1356
SOT101/0410D	24-Pin (300 mils wide) Plastic Dual In-Line (N) Package . . . . .	1357
SOT117	28-Pin Plastic Dual In-Line (N/P) Package . . . . .	1358
SOT129	40-Pin Plastic Dual In-Line (N/P) Package . . . . .	1359
SOT136A	28-Pin Plastic SO (Small Outline) Dual In-Line (D) Package . . . . .	1360
SOT158A	40-Pin Plastic VSO (Very Small Outline) Dual In-Line (D/T) Package . .	1361
SOT187	44-Pin Plastic Leaded Chip Carrier; Pocket Version (A) Package . . . . .	1362
SOT188AA	68-Pin Plastic Leaded Chip Carrier; Pocket Version (A) Package . . . . .	1363
SOT190	56-Pin Plastic VSO (Very Small Outline) Dual In-Line (D/T) Package . .	1364
SOT205	44-Pin Plastic Quad Flat Pack (B) Package . . . . .	1365
SOT261/0401F	28-Pin (300 mils wide) Plastic Leaded Chip Carrier (A) Package . . . . .	1366
SOT270AG	42-Lead Plastic Shrink Dual In-Line Package . . . . .	1367
SOT274	64-Pin Plastic Shrink Dual In-Line Package . . . . .	1368
SOT307-2	44-Pin Plastic Quad Flat Pack (B) Package . . . . .	1369
SOT311	44-Pin Square Plastic Quad Flat Pack (B) Package . . . . .	1370
SOT318	80-Pin Plastic Quad Flat Pack (B) Package . . . . .	1371
SOT319	64-Pin Plastic Quad Flat Pack Package . . . . .	1372
SOT340	24-Pin Plastic Shrink Small Outline Package (SSOP) . . . . .	1373
SOT341	28-Pin Plastic Shrink Small Outline Package (SSOP) . . . . .	1374
SOT351	80-Lead Ceramic Chip Carrier (CQFP) . . . . .	1375
SOT389	(refer to package number 1711—)	
0398E	68-Pin Plastic Leaded Chip Carrier (A) Package . . . . .	1376
0401F/SOT261	28-Pin (300 mils wide) Plastic Leaded Chip Carrier (A) Package . . . . .	1366
0403G	44-Pin Plastic Leaded Chip Carrier (A) Package . . . . .	1377
0410D/SOT101	24-Pin (300 mils wide) Plastic Dual In-Line (N) Package . . . . .	1357
0413B	28-pin (600 mils wide) Plastic Dual In-Line (N) Package . . . . .	1378
0414B	64-Pin (900 mils wide) Plastic Dual In-Line (N) Package . . . . .	1379
0415C	40-Pin (600 mils wide) Plastic Dual In-Line (N) Package . . . . .	1380
0586B	24-Pin (300 mils wide) Ceramic Dual In-line (F) Package (with Window (FA) Package) . . . . .	1381
0589B	28-Pin (600 mils wide) Ceramic Dual In-line (F) Package (with Window (FA) Package) . . . . .	1382
0590B	40-Pin (600 mils wide) Ceramic Dual In-line (F) Package (with Window (FA) Package) . . . . .	1383
1118D	44-Pin Plastic Quad Flat Pack (B) (MEC) Package . . . . .	1384
1240C	68-Pin Chip Carrier, J-Bend (L) Package . . . . .	1385
1472A	44-pin CerQuad J-Bend (K) Package . . . . .	1386
1473A	68-pin CerQuad J-Bend (K) Package . . . . .	1387
1680	42-Pin Plastic SSOP (Shrink Small Outline Package) Dual In-Line (D/K) Package . . . . .	1388
1711 —	44-Pin Plastic Thin Quad Flat Package (comparable to SOT389) . . . . .	1389





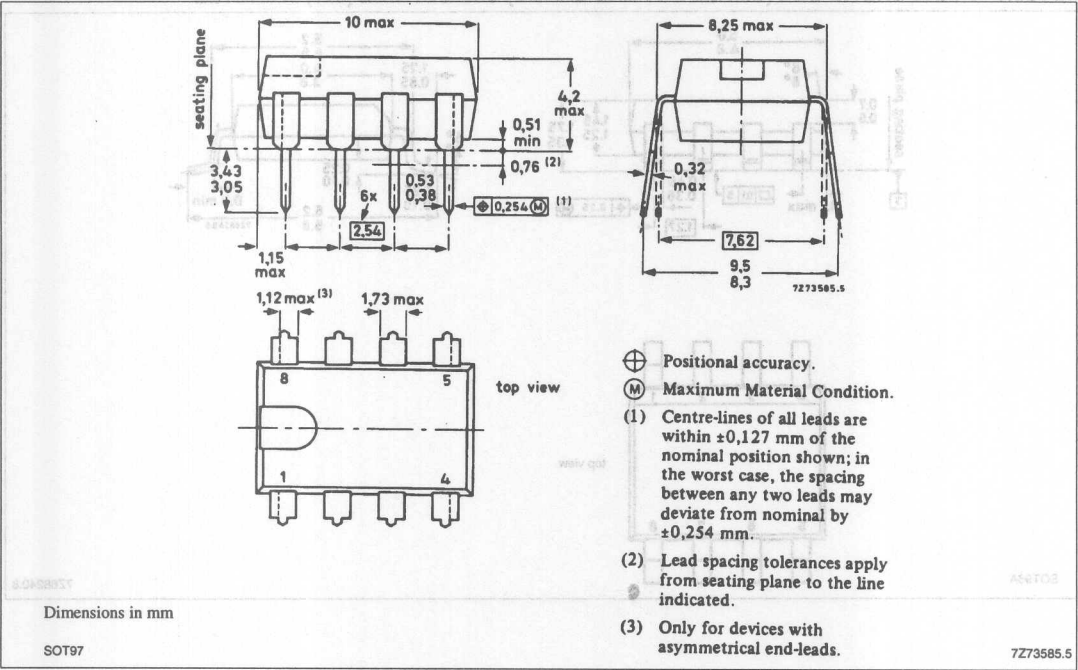
## Package outlines

### SOT96A 8-PIN PLASTIC SOL (SMALL OUTLINE LARGE) DUAL IN-LINE (D/T) PACKAGE



Package outlines

SOT97 8-PIN PLASTIC DUAL IN-LINE (N/P) PACKAGE

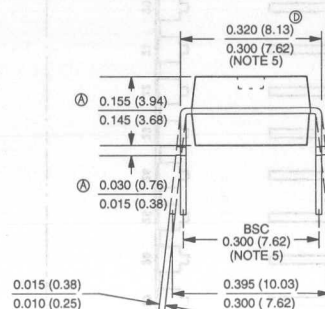
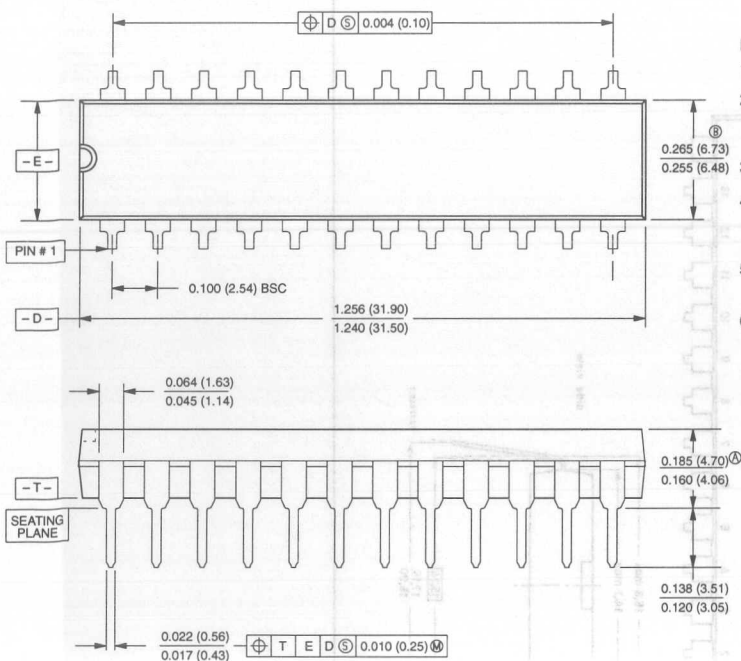


# Package outlines

## SOT101/0410D 24-PIN (300 mils wide) PLASTIC DUAL IN-LINE (N) PACKAGE

### NOTES:

1. Controlling dimension: Inches. Metric are shown in parentheses.
2. Package dimensions conform to JEDEC Specification MS-001-AF for standard Dual In-Line (DIP) package 0.300 inch row spacing (plastic) 24 leads (Issue B, 7/85).
3. Dimension and tolerancing per ANSI Y14, 5M - 1982.
4. "T", "D", and "E" are reference datums on the molded body and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.010 inch (0.25mm) on any side.
5. These dimensions measured with the leads constrained to be perpendicular to plane T.
6. Pin numbers start with Pin #1 and continue counterclockwise to Pin #24 when viewed from the top.

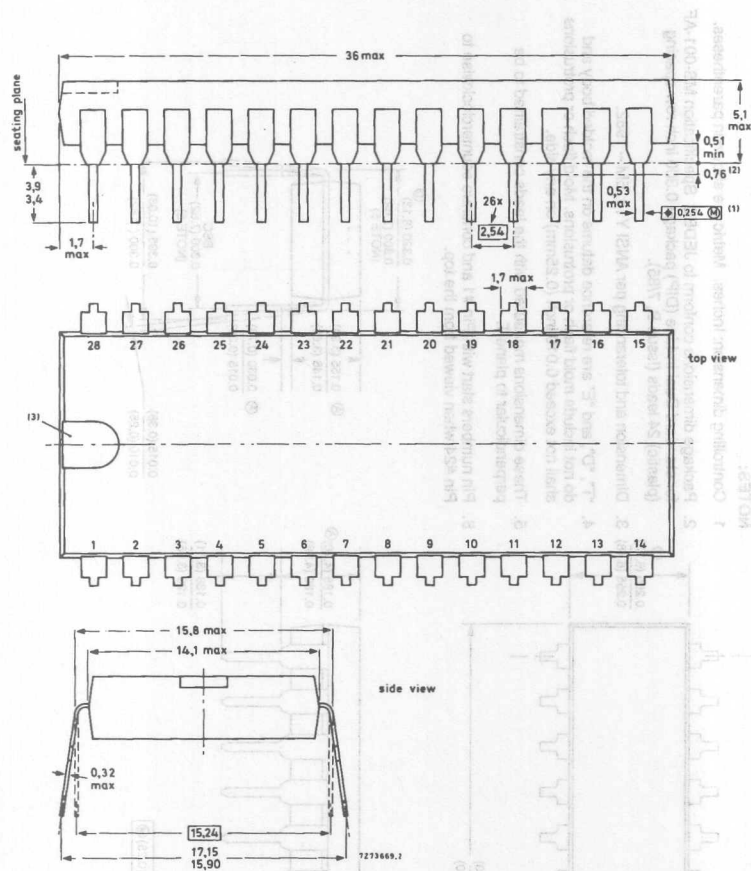


## Package outlines

## Package outlines

## SOT117 28-PIN PLASTIC DUAL IN-LINE (N/P) PACKAGE

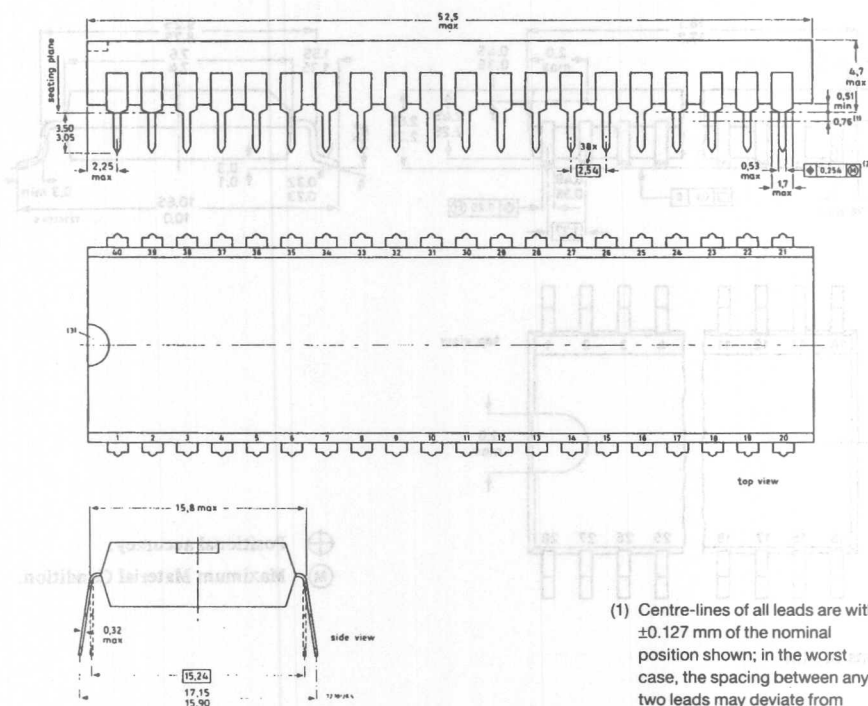
SOT117 (300 mils wide PLASTIC)



SOT117

## Package outlines

## SOT129 40-PIN PLASTIC DUAL IN-LINE (N/P) PACKAGE



(1) Centre-lines of all leads are within  $\pm 0.127$  mm of the nominal position shown; in the worst case, the spacing between any two leads may deviate from nominal by  $\pm 0.254$  mm.

(2) Lead spacing tolerances apply from seating plane to the line indicated.

(3) Index may be horizontal as shown, or vertical

(4) Dimensions in mm.

SOT 129

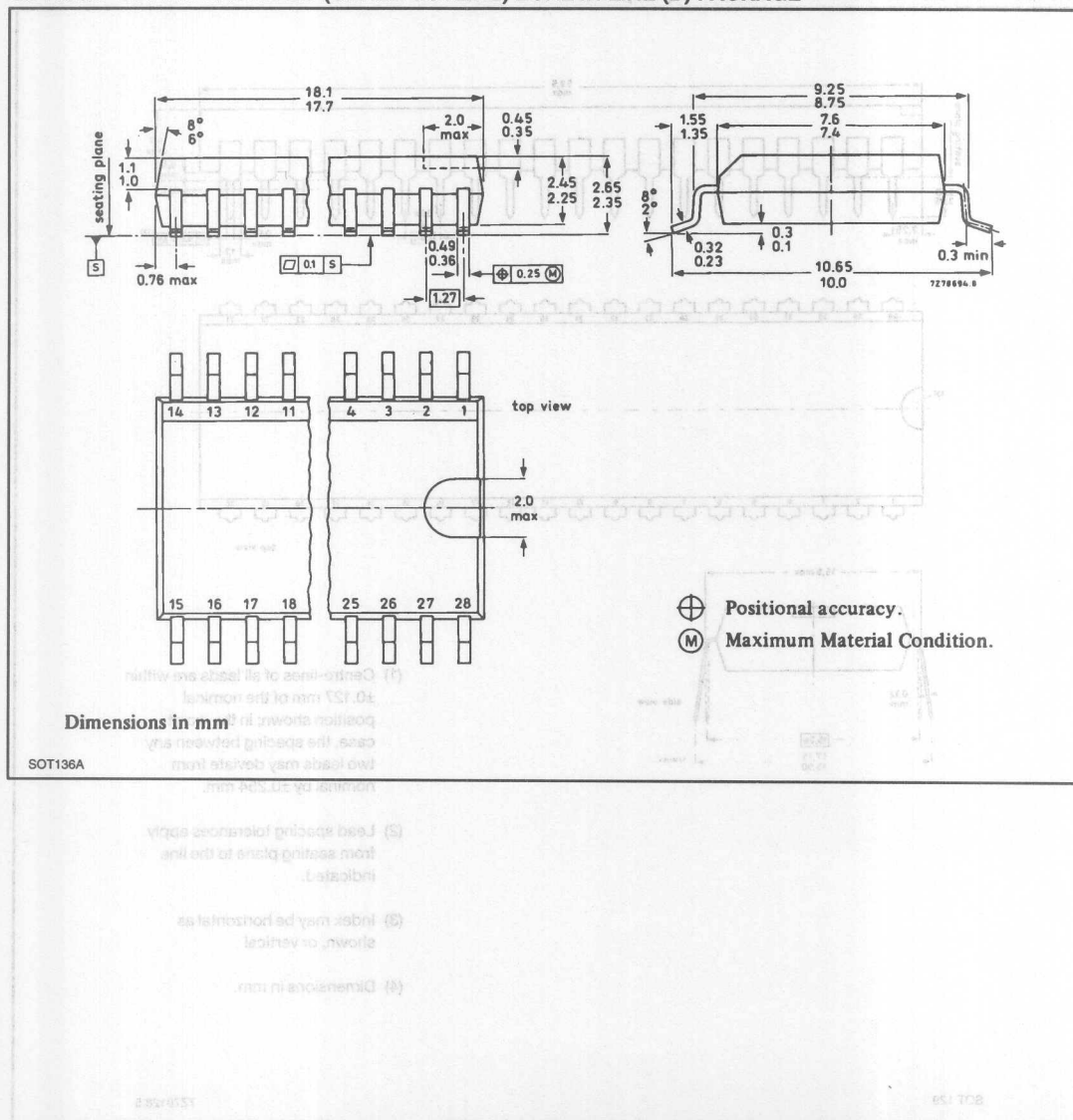
7Z70128.5



## Package outlines

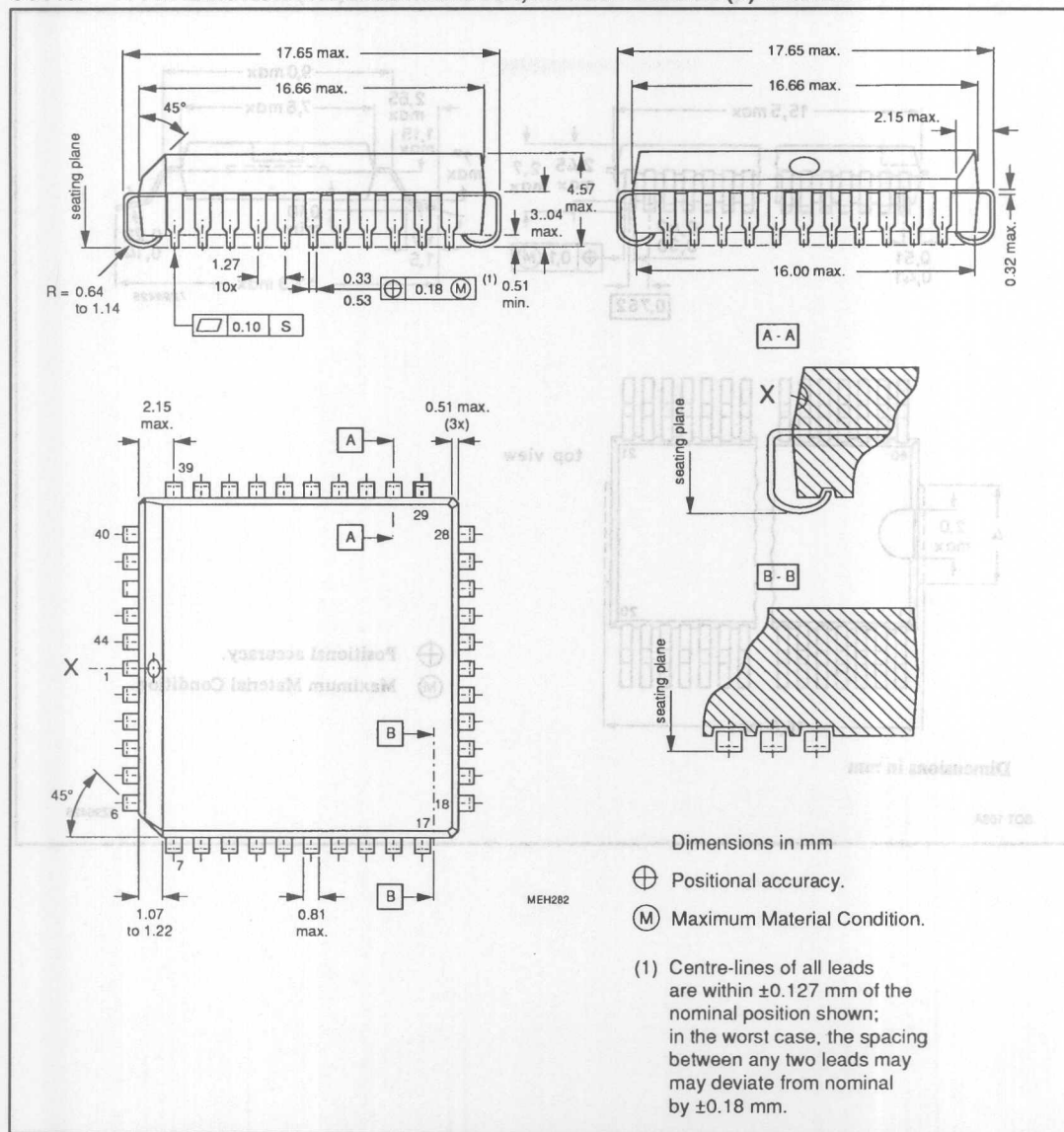
## Package outlines

## SOT136A 28-PIN PLASTIC SO (SMALL OUTLINE) DUAL IN-LINE (D) PACKAGE





# SOT187 44-PIN PLASTIC LEADED CHIP CARRIER; POCKET VERSION (A) PACKAGE

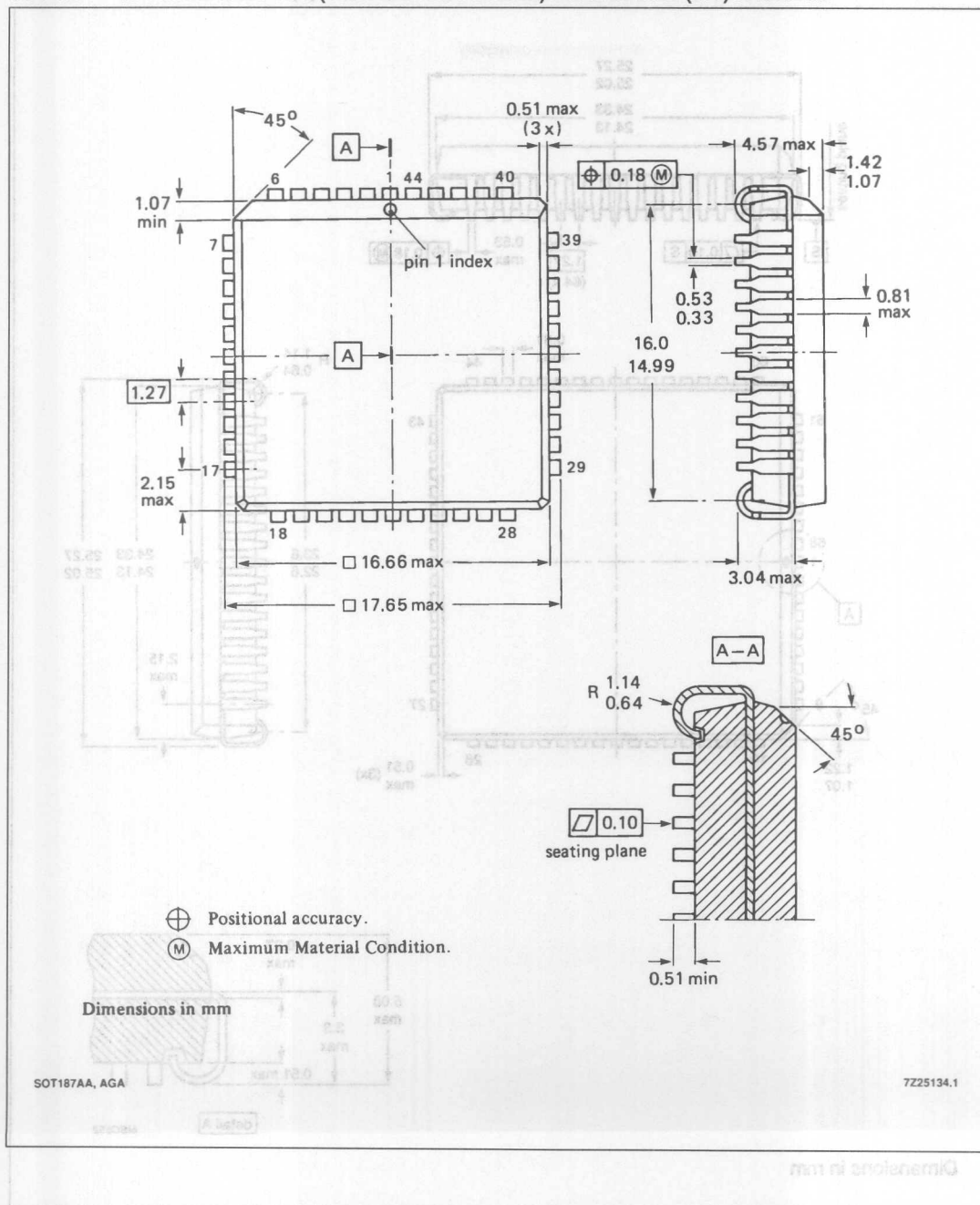




## Package outlines

## Package outlines

## SOT190 56-PIN PLASTIC VSO (VERY SMALL OUTLINE) DUAL IN-LINE (D/T) PACKAGE





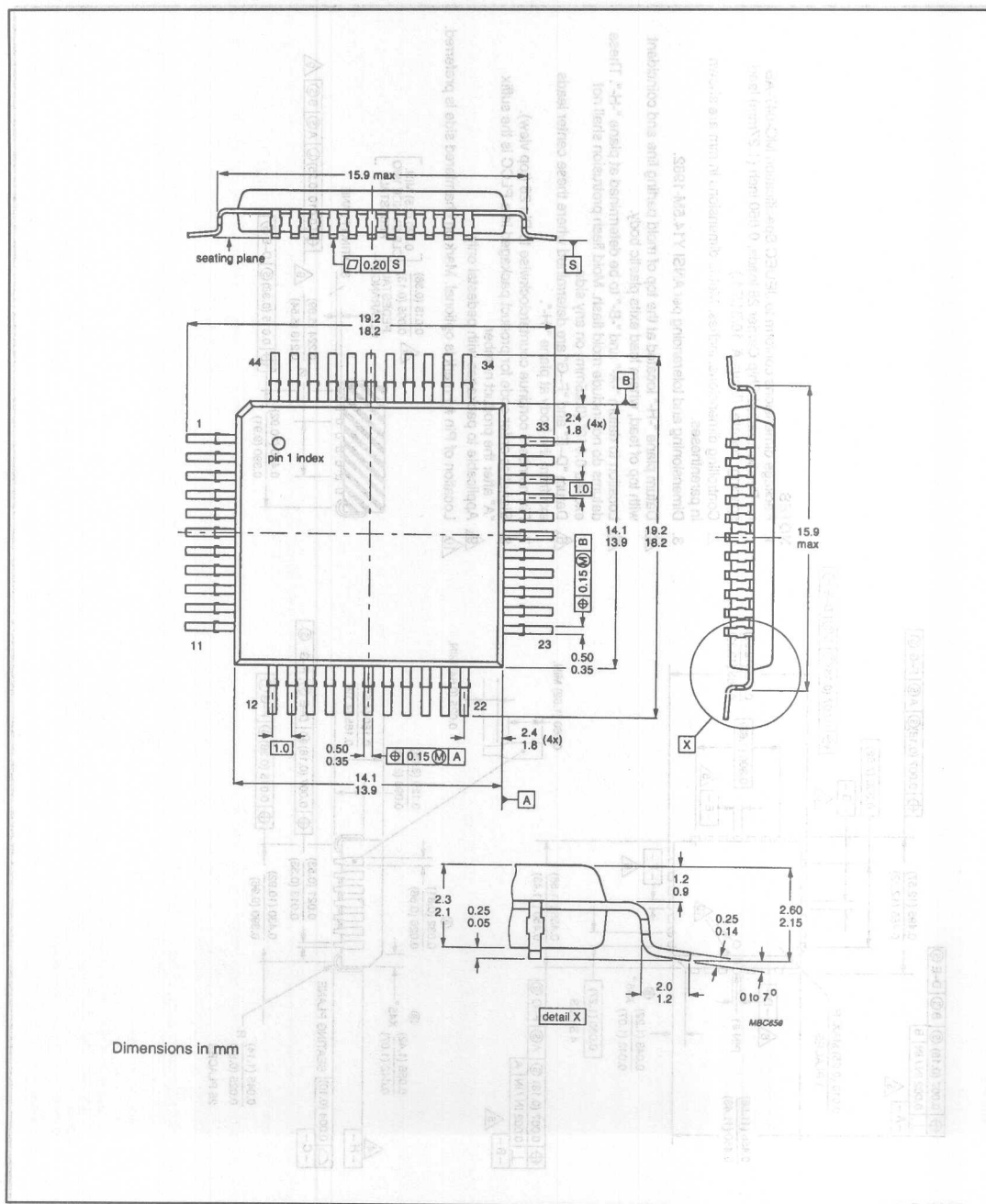
## Package outlines

## Package outlines

## SOT205 44-PIN PLASTIC QUAD FLAT PACK (B) PACKAGE

7.0-PIN (300 mils wide)

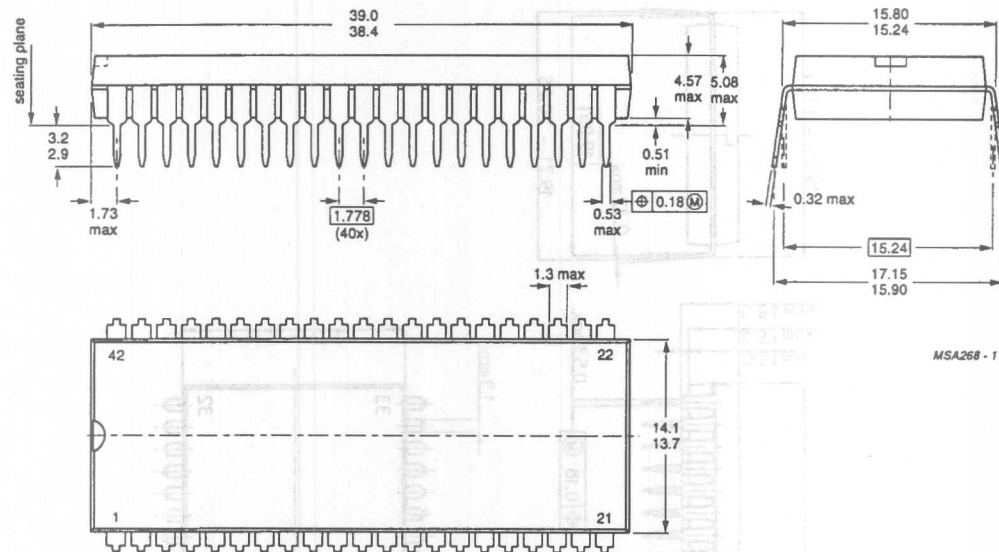
SOT205 44-PIN



- 
- 0.430 (10.92)
- 0.224 (5.69)
- 0.015 (0.38) MIN. CLEARANCE W/O PEDESTAL
- 0.015 (0.38) PEDESTAL CLEARANCE
- SEATING PLANE
- 0.010 (0.25) L A S B S
- 0.015 (0.38) S D-E S

## Package outlines

## SOT270AG 42-LEAD PLASTIC SHRINK DUAL IN-LINE PACKAGE



MSA268 - 1

Dimensions in mm.

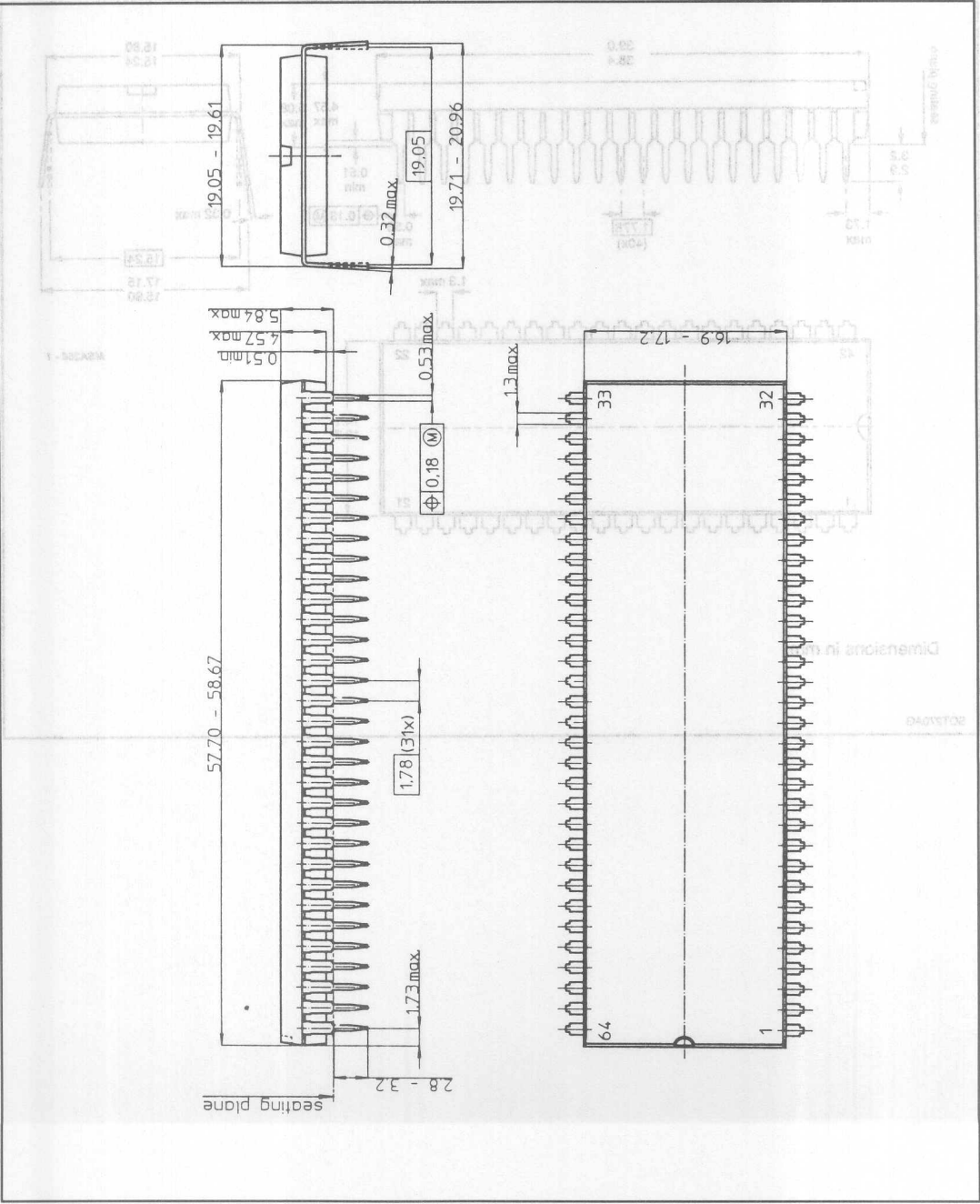
SOT270AG

Package outlines

Package outlines

SOT274 64-PIN PLASTIC SHRINK DUAL IN-LINE PACKAGE

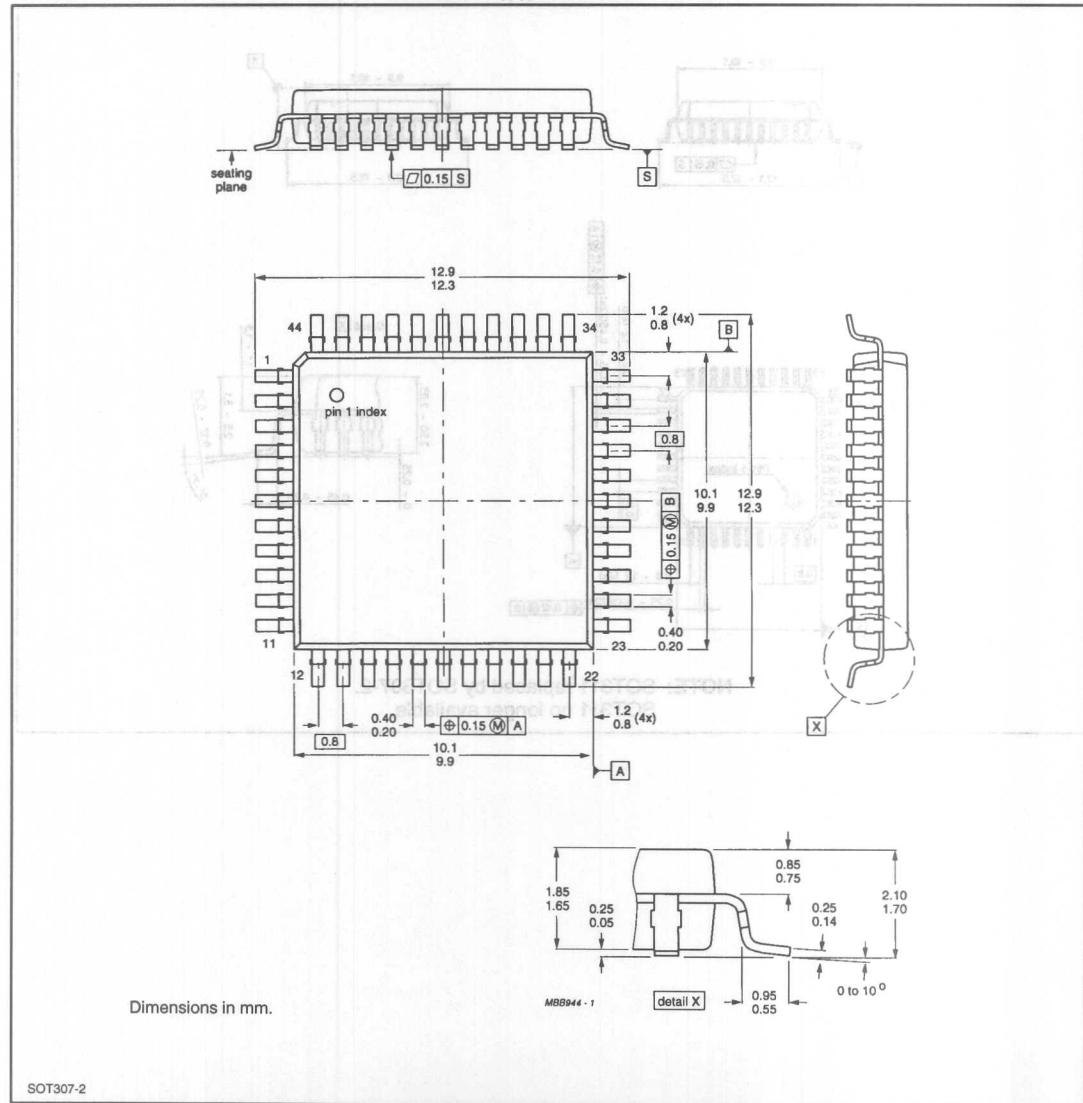
SOT274 64-LEAD PLASTIC SHRINK DUAL IN-LINE PACKAGE



## Package outlines

## Package outlines

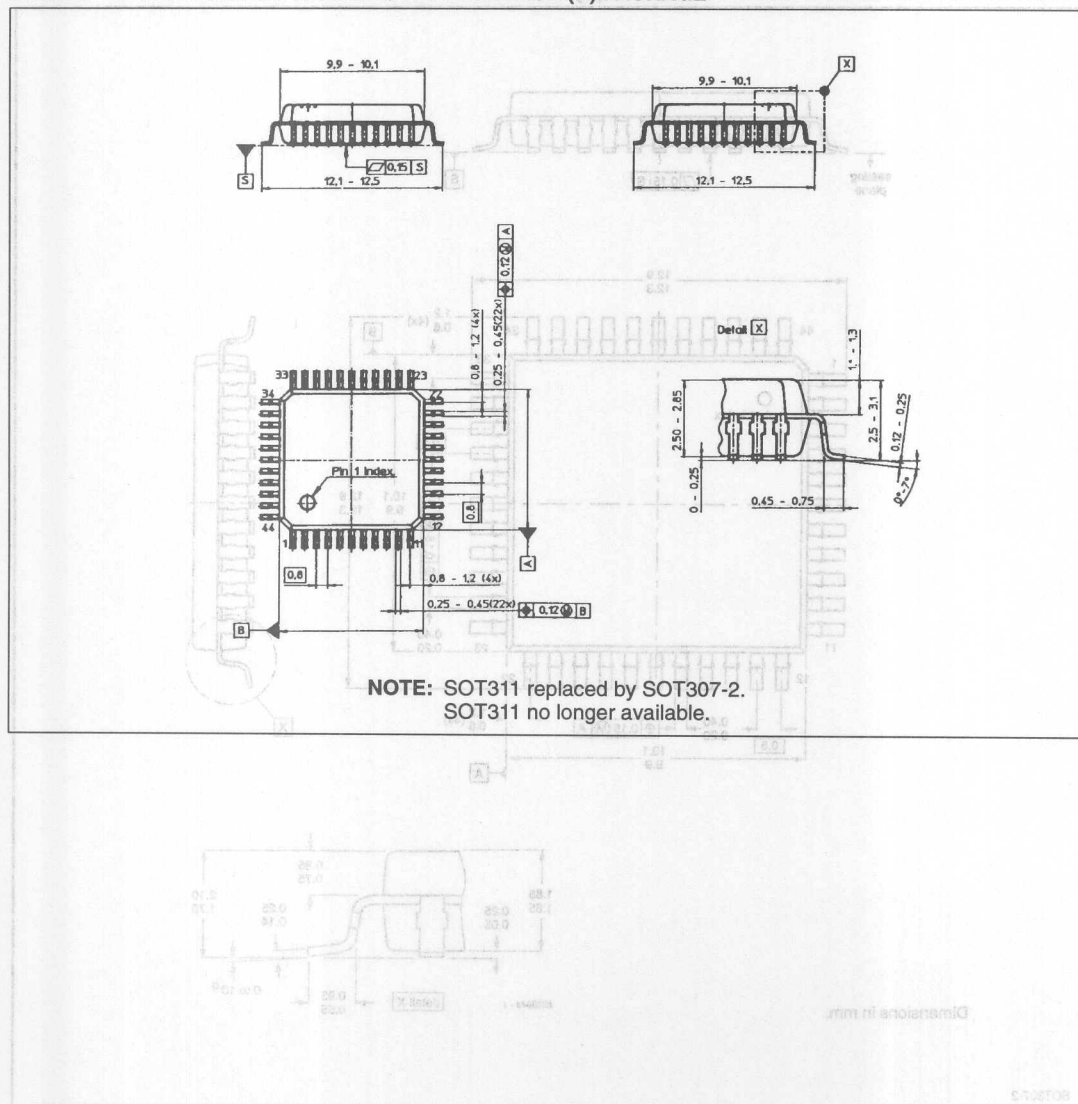
## SOT307-2 44-PIN PLASTIC QUAD FLAT PACK (B) PACKAGE





## Package outlines

**SOT311 44-PIN SQUARE PLASTIC QUAD FLAT PACK (B) PACKAGE**

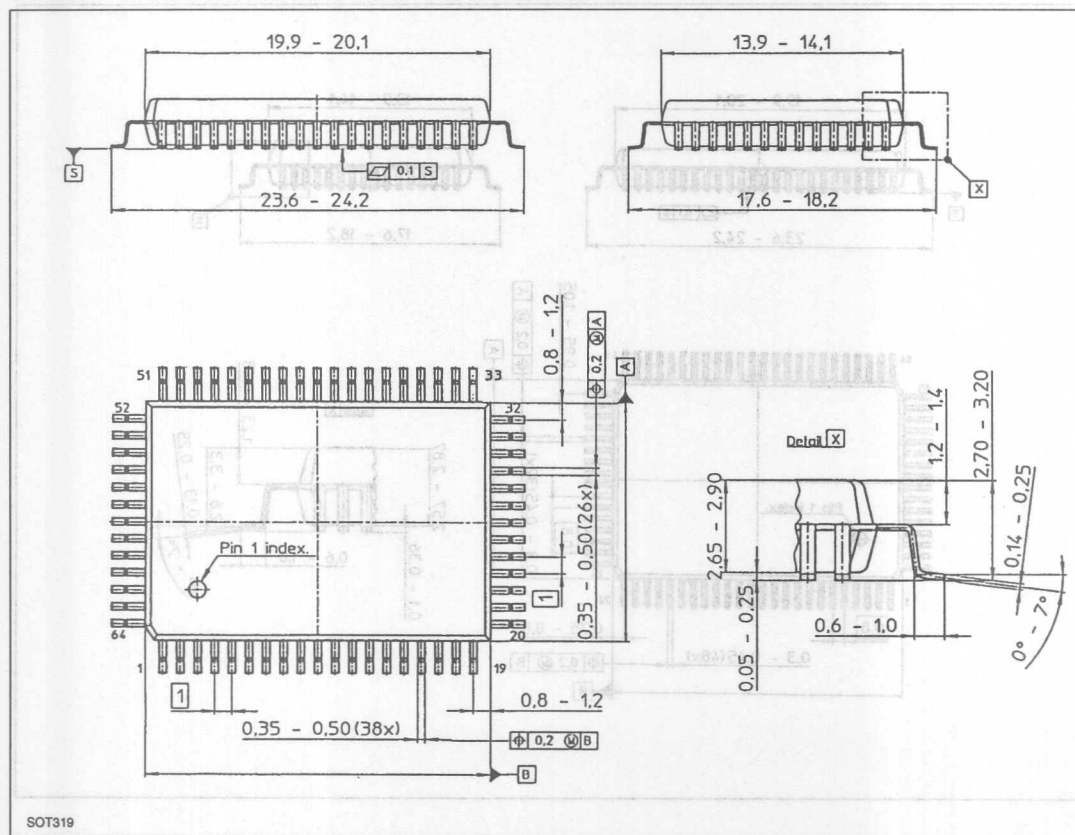




## Package outlines

## Package outlines

## SOT319 64-PIN PLASTIC QUAD FLAT PACK PACKAGE

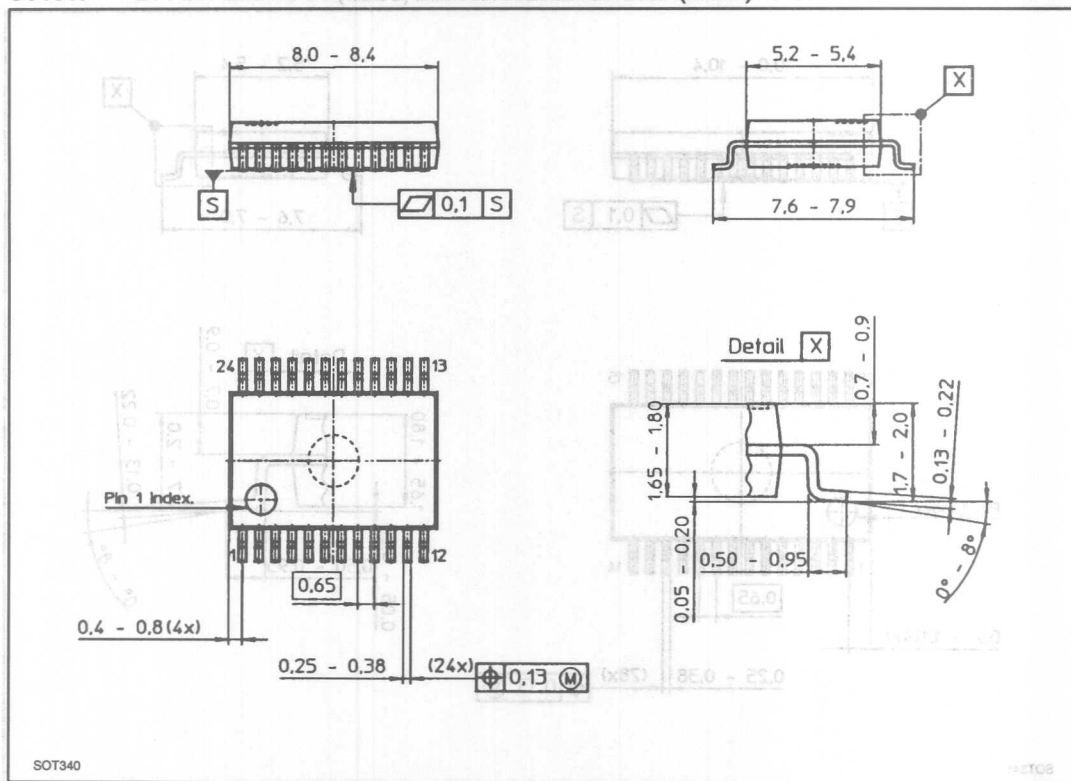


## Package outlines

## Package outlines

## SOT340 24-PIN PLASTIC SHRINK SMALL OUTLINE PACKAGE (SSOP)

SOT340

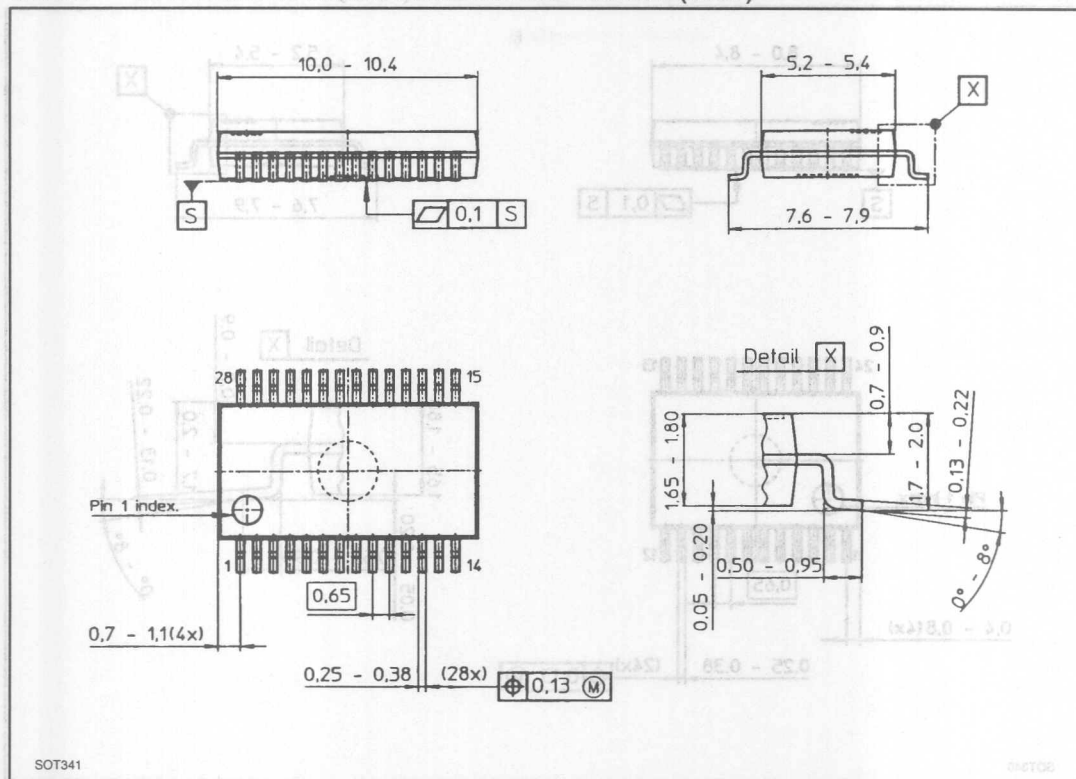


## Package outlines

## Package outlines

SOT341 28-PIN PLASTIC SHRINK SMALL OUTLINE PACKAGE (SSOP)

SOT341





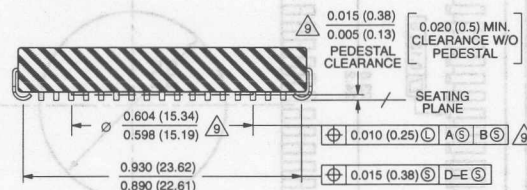
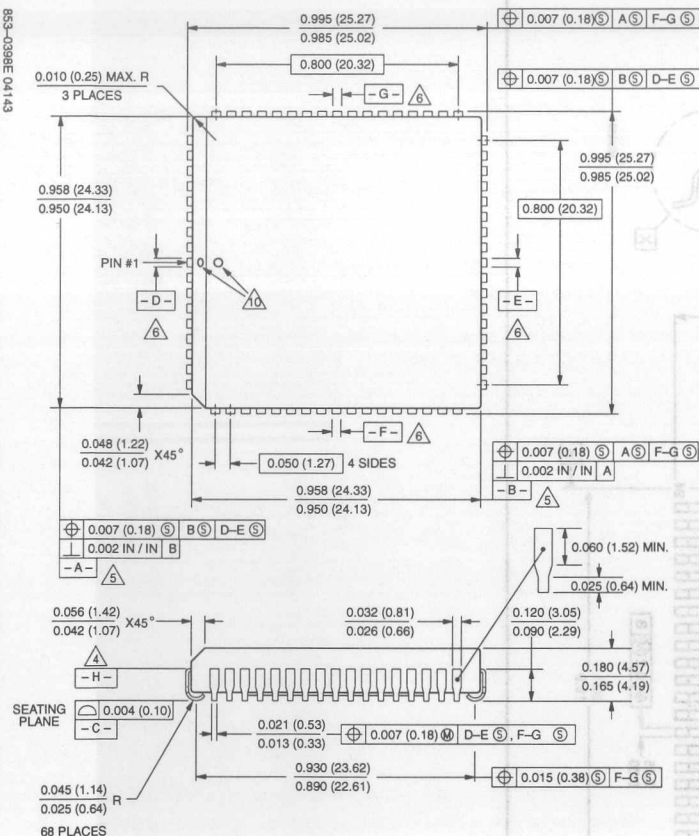


# Package outlines

## 0388E 68-PIN PLASTIC LEADED CHIP CARRIER (A) PACKAGE

### NOTES

- Package dimensions conform to JEDEC Specification MO-047-AE for Plastic Leaded Chip Carrier 68 leads, 0.050 inch lead spacing, square. (Issue A, 10/31/84).
- Controlling dimensions: inches. Metric dimensions in mm are shown in parentheses.
- Dimensioning and tolerancing per ANSI Y14.5M-1982.
- Datum plane "H" located at the top of mold parting line and coincident with top of lead, where lead exits plastic body.
- Location to datum "A" and "B" to be determined at plane "H". These datums do not include mold flash. Mold flash protrusion shall not exceed 0.010" (0.25mm) on any side.
- Datum "D-E" and "F-G" are determined where these center leads exit from the body at plane "H".
- Pin numbers continue counterclockwise to Pin 68 (top view).
- Signetics order code for product packaged in a PLCC is the suffix "A" after the product number.
- Applicable to packages with pedestal only.
- Location of Pin #1 mark is optional. Mark on chamfered side is preferred.

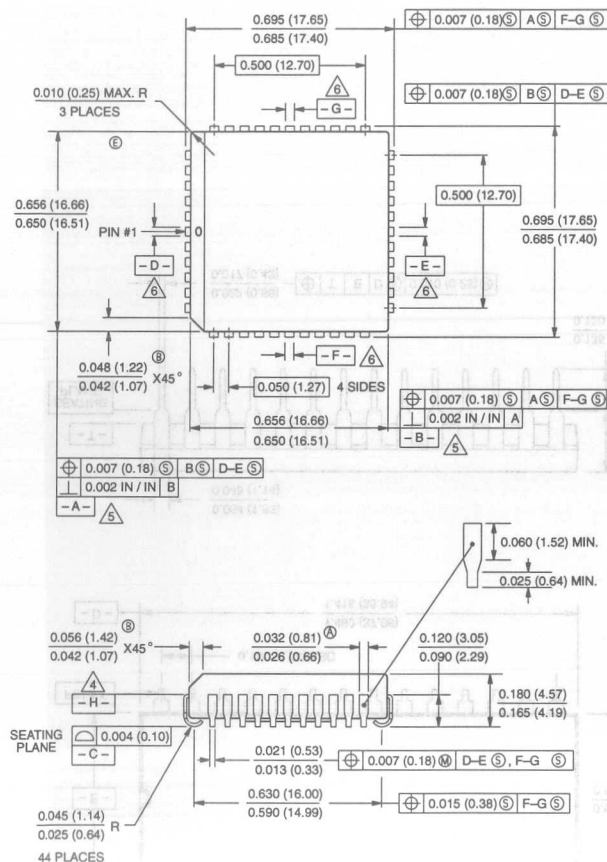
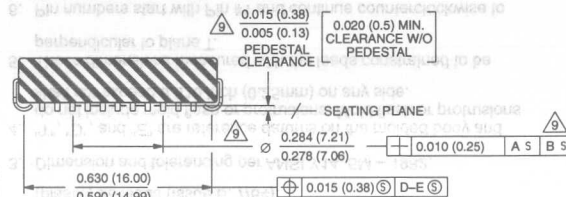


## Package outlines

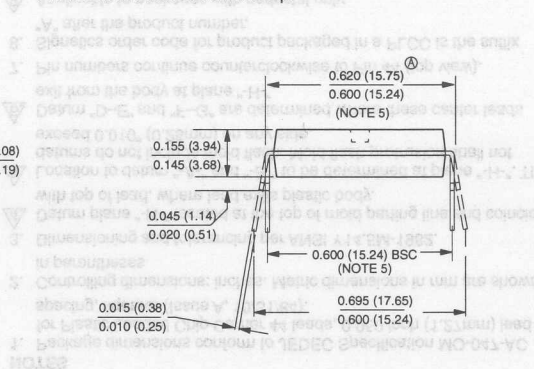
0403G 44-PIN PLASTIC LEADED CHIP CARRIER (A) PACKAGE T2A, T2B, T2C, T2D, T2E, T2F, T2G, T2H, T2I, T2J, T2K, T2L, T2M, T2N, T2O, T2P, T2Q, T2R, T2S, T2T, T2U, T2V, T2W, T2X, T2Y, T2Z, T2AA, T2AB, T2AC, T2AD, T2AE, T2AF, T2AG, T2AH, T2AI, T2AJ, T2AK, T2AL, T2AM, T2AN, T2AO, T2AP, T2AQ, T2AR, T2AS, T2AT, T2AU, T2AV, T2AW, T2AX, T2AY, T2AZ, T2BA, T2BB, T2BC, T2BD, T2BE, T2BF, T2BG, T2BH, T2BI, T2BJ, T2BK, T2BL, T2BM, T2BN, T2BO, T2BP, T2BQ, T2BR, T2BS, T2BT, T2BU, T2BV, T2BW, T2BX, T2BY, T2BZ, T2CA, T2CB, T2CC, T2CD, T2CE, T2CF, T2CG, T2CH, T2CI, T2CJ, T2CK, T2CL, T2CM, T2CN, T2CO, T2CP, T2CQ, T2CR, T2CS, T2CT, T2CU, T2CV, T2CW, T2CX, T2CY, T2CZ, T2DA, T2DB, T2DC, T2DD, T2DE, T2DF, T2DG, T2DH, T2DI, T2DJ, T2DK, T2DL, T2DM, T2DN, T2DO, T2DP, T2DQ, T2DR, T2DS, T2DT, T2DU, T2DV, T2DW, T2DX, T2DY, T2DZ, T2EA, T2EB, T2EC, T2ED, T2EE, T2EF, T2EG, T2EH, T2EI, T2EJ, T2EK, T2EL, T2EM, T2EN, T2EO, T2EP, T2EQ, T2ER, T2ES, T2ET, T2EU, T2EV, T2EW, T2EX, T2EY, T2EZ, T2FA, T2FB, T2FC, T2FD, T2FE, T2FF, T2FG, T2FH, T2FI, T2FJ, T2FK, T2FL, T2FM, T2FN, T2FO, T2FP, T2FQ, T2FR, T2FS, T2FT, T2FU, T2FV, T2FW, T2FX, T2FY, T2FZ, T2GA, T2GB, T2GC, T2GD, T2GE, T2GF, T2GG, T2GH, T2GI, T2GJ, T2GK, T2GL, T2GM, T2GN, T2GO, T2GP, T2GQ, T2GR, T2GS, T2GT, T2GU, T2GV, T2GW, T2GX, T2GY, T2GZ, T2HA, T2HB, T2HC, T2HD, T2HE, T2HF, T2HG, T2HH, T2HI, T2HJ, T2HK, T2HL, T2HM, T2HN, T2HO, T2HP, T2HQ, T2HR, T2HS, T2HT, T2HU, T2HV, T2HW, T2HX, T2HY, T2HZ, T2IA, T2IB, T2IC, T2ID, T2IE, T2IF, T2IG, T2IH, T2II, T2IJ, T2IK, T2IL, T2IM, T2IN, T2IO, T2IP, T2IQ, T2IR, T2IS, T2IT, T2IU, T2IV, T2IW, T2IX, T2IY, T2IZ, T2JA, T2JB, T2JC, T2JD, T2JE, T2JF, T2JG, T2JH, T2JI, T2JJ, T2JK, T2JL, T2JM, T2JN, T2JO, T2JP, T2JQ, T2JR, T2JS, T2JT, T2JU, T2JV, T2JW, T2JX, T2JY, T2JZ, T2KA, T2KB, T2KC, T2KD, T2KE, T2KF, T2KG, T2KH, T2KI, T2KJ, T2KK, T2KL, T2KM, T2KN, T2KO, T2KP, T2KQ, T2KR, T2KS, T2KT, T2KU, T2KV, T2KW, T2KX, T2KY, T2KZ, T2LA, T2LB, T2LC, T2LD, T2LE, T2LF, T2LG, T2LH, T2LI, T2LJ, T2LK, T2LL, T2LM, T2LN, T2LO, T2LP, T2LQ, T2LR, T2LS, T2LT, T2LU, T2LV, T2LW, T2LX, T2LY, T2LZ, T2MA, T2MB, T2MC, T2MD, T2ME, T2MF, T2MG, T2MH, T2MI, T2MJ, T2MK, T2ML, T2MM, T2MN, T2MO, T2MP, T2MQ, T2MR, T2MS, T2MT, T2MU, T2MV, T2MW, T2MX, T2MY, T2MZ, T2NA, T2NB, T2NC, T2ND, T2NE, T2NF, T2NG, T2NH, T2NI, T2NJ, T2NK, T2NL, T2NM, T2NN, T2NO, T2NP, T2NQ, T2NR, T2NS, T2NT, T2NU, T2NV, T2NW, T2NX, T2NY, T2NZ, T2OA, T2OB, T2OC, T2OD, T2OE, T2OF, T2OG, T2OH, T2OI, T2OJ, T2OK, T2OL, T2OM, T2ON, T2OO, T2OP, T2OQ, T2OR, T2OS, T2OT, T2OU, T2OV, T2OW, T2OX, T2OY, T2OZ, T2PA, T2PB, T2PC, T2PD, T2PE, T2PF, T2PG, T2PH, T2PI, T2PJ, T2PK, T2PL, T2PM, T2PN, T2PO, T2PP, T2PQ, T2PR, T2PS, T2PT, T2PU, T2PV, T2PW, T2PX, T2PY, T2PZ, T2QA, T2QB, T2QC, T2QD, T2QE, T2QF, T2QG, T2QH, T2QI, T2QJ, T2QK, T2QL, T2QM, T2QN, T2QO, T2QP, T2QQ, T2QR, T2QS, T2QT, T2QU, T2QV, T2QW, T2QX, T2QY, T2QZ, T2RA, T2RB, T2RC, T2RD, T2RE, T2RF, T2RG, T2RH, T2RI, T2RJ, T2RK, T2RL, T2RM, T2RN, T2RO, T2RP, T2RQ, T2RR, T2RS, T2RT, T2RU, T2RV, T2RW, T2RX, T2RY, T2RZ, T2SA, T2SB, T2SC, T2SD, T2SE, T2SF, T2SG, T2SH, T2SI, T2SJ, T2SK, T2SL, T2SM, T2SN, T2SO, T2SP, T2SQ, T2SR, T2SS, T2ST, T2SU, T2SV, T2SW, T2SX, T2SY, T2SZ, T2TA, T2TB, T2TC, T2TD, T2TE, T2TF, T2TG, T2TH, T2TI, T2TJ, T2TK, T2TL, T2TM, T2TN, T2TO, T2TP, T2TQ, T2TR, T2TS, T2TT, T2TU, T2TV, T2TW, T2TX, T2TY, T2TZ, T2UA, T2UB, T2UC, T2UD, T2UE, T2UF, T2UG, T2UH, T2UI, T2UJ, T2UK, T2UL, T2UM, T2UN, T2UO, T2UP, T2UQ, T2UR, T2US, T2UT, T2UU, T2UV, T2UW, T2UX, T2UY, T2UZ, T2VA, T2VB, T2VC, T2VD, T2VE, T2VF, T2VG, T2VH, T2VI, T2VJ, T2VK, T2VL, T2VM, T2VN, T2VO, T2VP, T2VQ, T2VR, T2VS, T2VT, T2VU, T2VV, T2VW, T2VX, T2VY, T2VZ, T2WA, T2WB, T2WC, T2WD, T2WE, T2WF, T2WG, T2WH, T2WI, T2WJ, T2WK, T2WL, T2WM, T2WN, T2WO, T2WP, T2WQ, T2WR, T2WS, T2WT, T2WU, T2WV, T2WW, T2WX, T2WY, T2WZ, T2XA, T2XB, T2XC, T2XD, T2XE, T2XF, T2XG, T2XH, T2XI, T2XJ, T2XK, T2XL, T2XM, T2XN, T2XO, T2XP, T2XQ, T2XR, T2XS, T2XT, T2XU, T2XV, T2XW, T2XX, T2XY, T2XZ, T2YA, T2YB, T2YC, T2YD, T2YE, T2YF, T2YG, T2YH, T2YI, T2YJ, T2YK, T2YL, T2YM, T2YN, T2YO, T2YP, T2YQ, T2YR, T2YS, T2YT, T2YU, T2YV, T2YW, T2YX, T2YY, T2YZ, T2ZA, T2ZB, T2ZC, T2ZD, T2ZE, T2ZF, T2ZG, T2ZH, T2ZI, T2ZJ, T2ZK, T2ZL, T2ZM, T2ZN, T2ZO, T2ZP, T2ZQ, T2ZR, T2ZS, T2ZT, T2ZU, T2ZV, T2ZW, T2ZX, T2ZY, T2ZZ

## NOTES

- Package dimensions conform to JEDEC Specification MO-047-AC for Plastic Leaded Chip Carrier 44 leads, 0.050 inch (1.27mm) lead spacing, square. (Issue A, 10/31/84).
- Controlling dimensions: inches. Metric dimensions in mm are shown in parentheses.
- Dimensioning and tolerancing per ANSI Y14.5M-1982.
- Datum plane "H-" located at the top of mold parting line and coincident with top of lead, where lead exits plastic body.
- Location to datum "A-" and "B-" to be determined at plane "H-". These datums do not include mold flash. Mold flash protrusion shall not exceed 0.010" (0.25mm) on any side.
- Datum "D-E" and "F-G" are determined where these center leads exit from the body at plane "H-".
- Pin numbers continue counterclockwise to Pin 44 (top view).
- Signetics order code for product packaged in a PLCC is the suffix "A" after the product number.
- Applicable to packages with pedestal only.

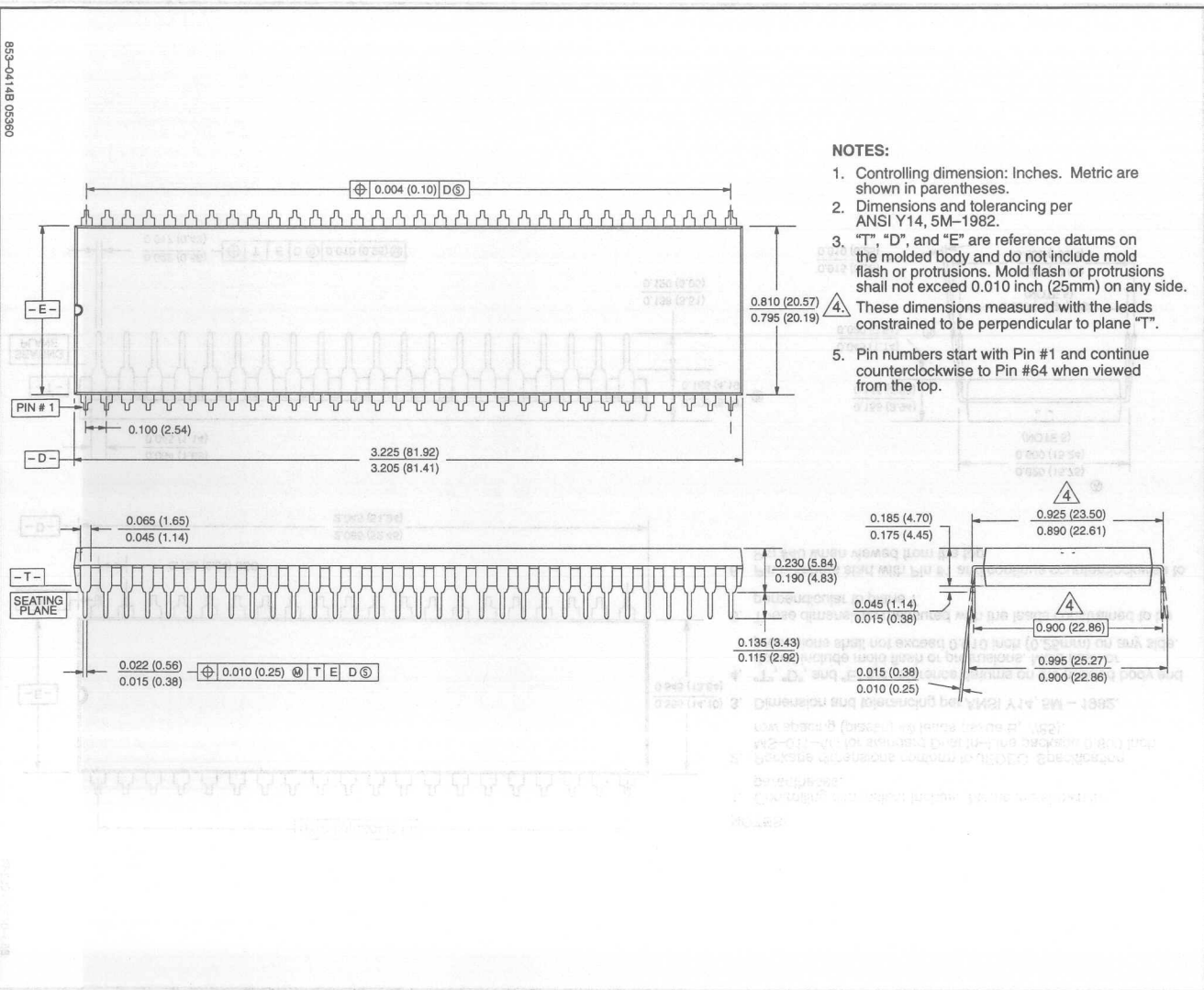


PCOM



# Package outlines

## 0414B 64-PIN (900 mils wide) PLASTIC DUAL IN-LINE (N) PACKAGE





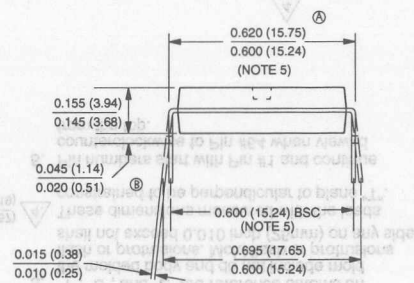
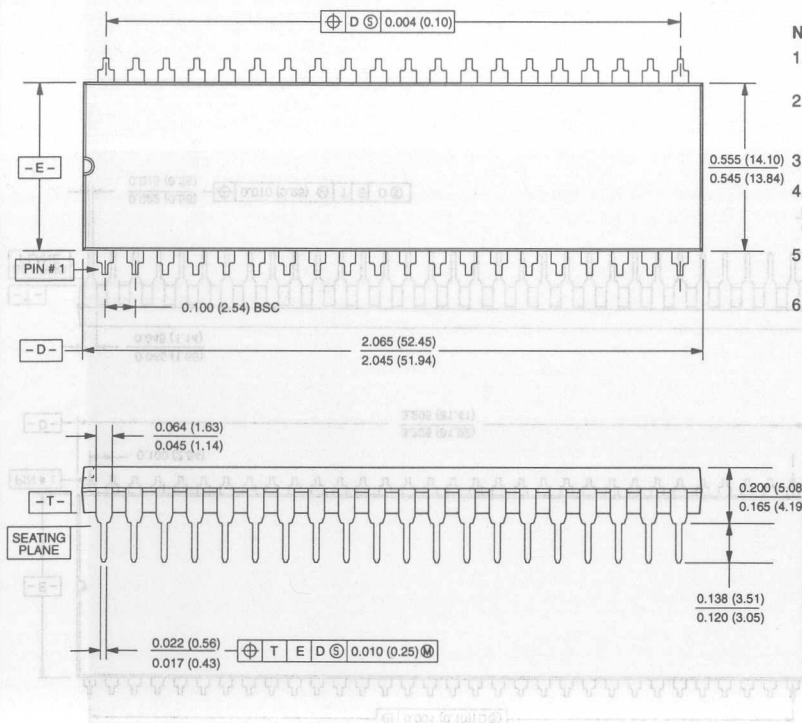
# Package outlines

80C51-Based 8-Bit Microcontrollers

## 0415C 40-PIN (600 mils wide) PLASTIC DUAL IN-LINE (N) PACKAGE

### NOTES:

1. Controlling dimension: Inches. Metric are shown in parentheses.
2. Package dimensions conform to JEDEC Specification MS-011-AC for standard Dual In-Line package 0.600 inch row spacing (plastic) 40 leads (Issue B, 7/85).
3. Dimension and tolerancing per ANSI Y14, 5M - 1982.
4. "T", "D", and "E" are reference datums on the molded body and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.010 inch (0.25mm) on any side.
5. These dimensions measured with the leads constrained to be perpendicular to plane T.
6. Pin numbers start with Pin #1 and continue counterclockwise to Pin #40 when viewed from the top.



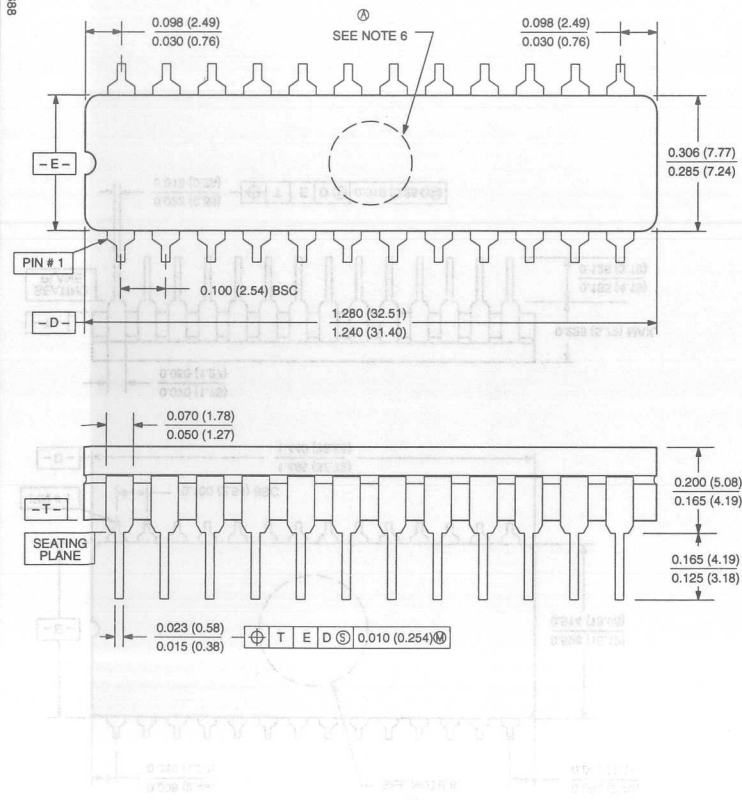
NOTES:

1. Controlling dimension: inches. Metric are shown in parentheses.
2. Package dimensions conform to JEDEC Specification MS-011-AC for standard Dual In-Line package 0.600 inch row spacing (plastic) 40 leads (Issue B, 7/85).
3. Dimension and tolerancing per ANSI Y14, 5M - 1982.
4. "T", "D", and "E" are reference datums on the molded body and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.010 inch (0.25mm) on any side.
5. These dimensions measured with the leads constrained to be perpendicular to plane T.
6. Pin numbers start with Pin #1 and continue counterclockwise to Pin #40 when viewed from the top.

# Package outlines

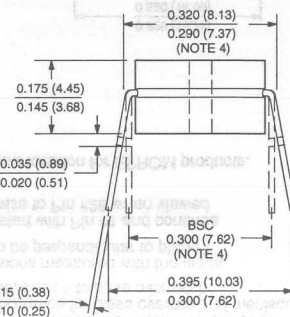
## 0586B 24-PIN (300 mils wide) CERAMIC DUAL IN-LINE (F) PACKAGE (WITH WINDOW (FA) PACKAGE)

853-0586B 06698



### NOTES:

1. Controlling dimension: Inches. Millimeters are shown in parentheses.
2. Dimension and tolerancing per ANSI Y14. 5M-1982.
3. "T", "D", and "E" are reference datums on the body and include allowance for glass overrun and meniscus on the seal line, and lid to base mismatch.
4. These dimensions measured with the leads constrained to be perpendicular to plane T.
5. Pin numbers start with Pin #1 and continue counterclockwise to Pin #24 when viewed from the top.
6. Denotes window location for EPROM products.



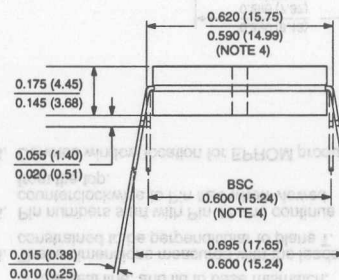
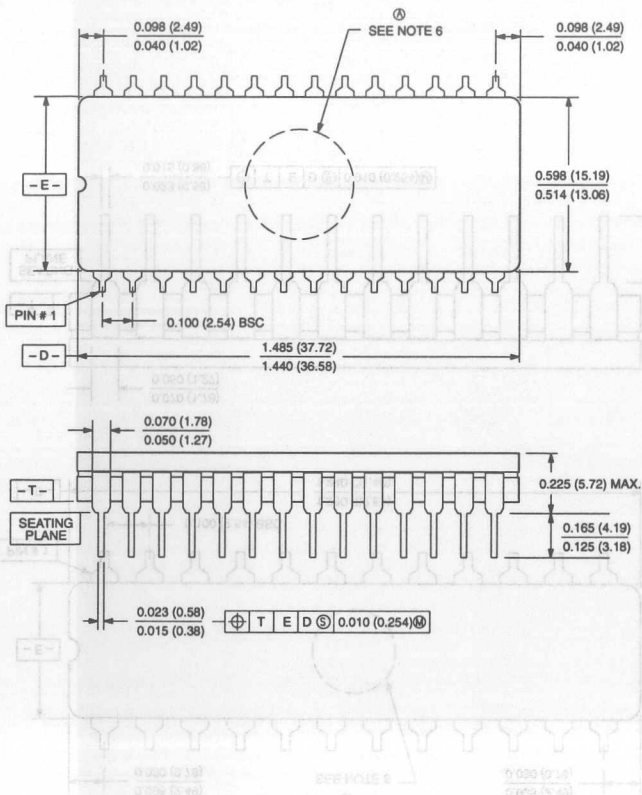
# Package outlines

0589B 28-PIN (600 mils wide) CERAMIC DUAL IN-LINE (F) PACKAGE (WITH WINDOW (FA) PACKAGE)

## NOTES:

1. Controlling dimension: Inches. Millimeters are shown in parentheses.
2. Dimension and tolerancing per ANSI Y14. 5M-1982.
3. "T", "D", and "E" are reference datums on the body and include allowance for glass overrun and meniscus on the seal line, and lid to base mismatch.
4. These dimensions measured with the leads constrained to be perpendicular to plane T.
5. Pin numbers start with Pin #1 and continue counterclockwise to Pin #28 when viewed from the top.

⑥ 6. Denotes window location for EPROM products.



## NOTES:

1. Controlling dimension: inches. Millimeters are shown in parentheses.

2. Dimension and tolerancing per ANSI Y14. 5M-1982.

3. "T", "D", and "E" are reference datums on the body and include allowance for glass overrun and meniscus on the seal line, and lid to base mismatch.

4. These dimensions measured with the leads constrained to be perpendicular to plane T.

5. Pin numbers start with Pin #1 and continue counterclockwise to Pin #28 when viewed from the top.

6. Denotes window location for EPROM products.

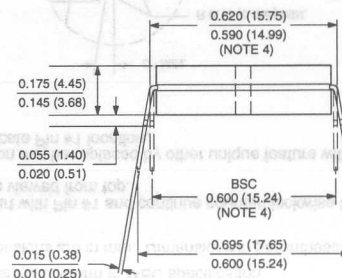
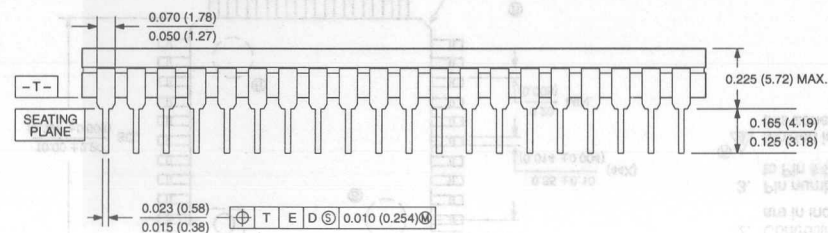
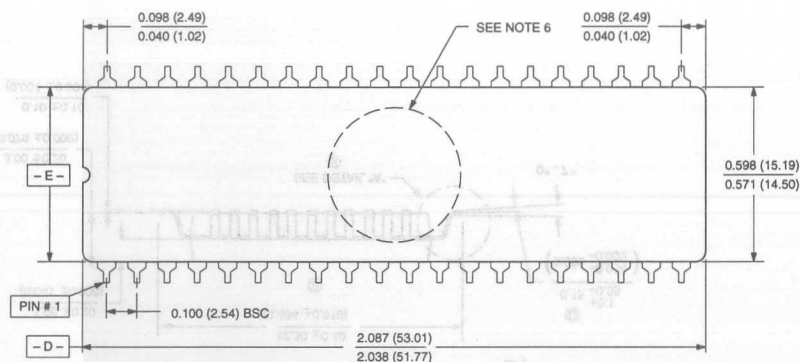
0589B 28-PIN (600 mils wide) CERAMIC DUAL IN-LINE (F) PACKAGE (WITH WINDOW (FA) PACKAGE)

# Package outlines

0590B 40-PIN (600 mils wide) CERAMIC DUAL IN-LINE (F) PACKAGE (WITH WINDOW (FA) PACKAGE)

## NOTES:

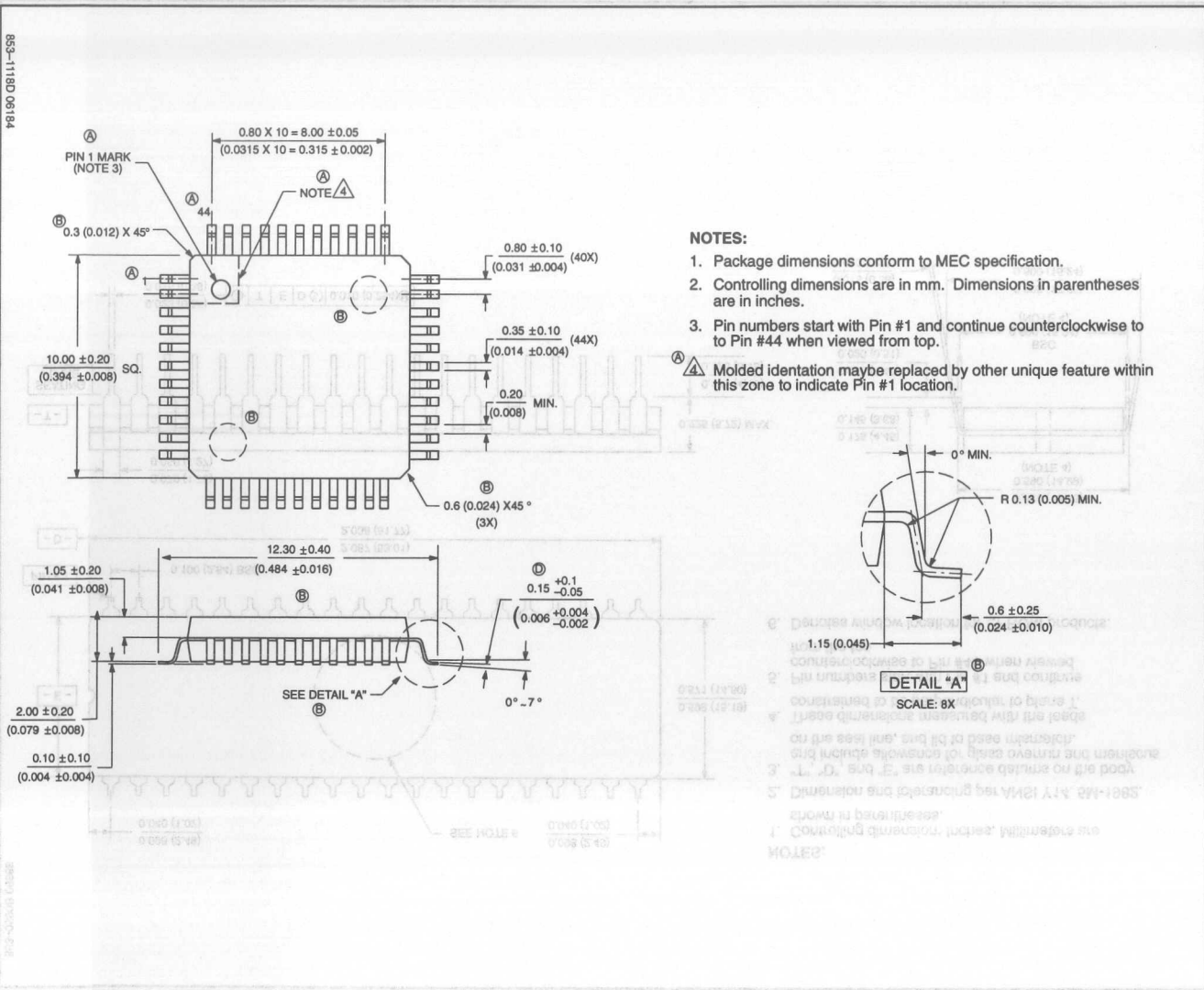
1. Controlling dimension: Inches. Millimeters are shown in parentheses.
2. Dimension and tolerancing per ANSI Y14. 5M-1982.
3. "T", "D", and "E" are reference datums on the body and include allowance for glass overrun and meniscus on the seal line, and lid to base mismatch.
4. These dimensions measured with the leads constrained to be perpendicular to plane T.
5. Pin numbers start with Pin #1 and continue counterclockwise to Pin #40 when viewed from the top.
6. Denotes window location for EPROM products.



# Package outlines

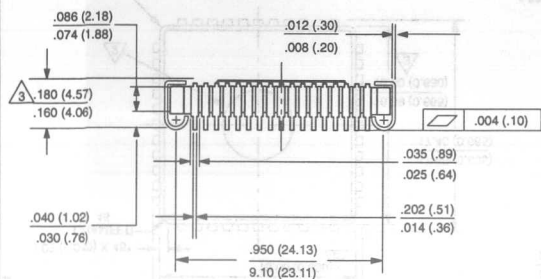
## Package outlines

### 1118D 44-PIN PLASTIC QUAD FLAT PACK (B) (MEC) PACKAGE





ASTM



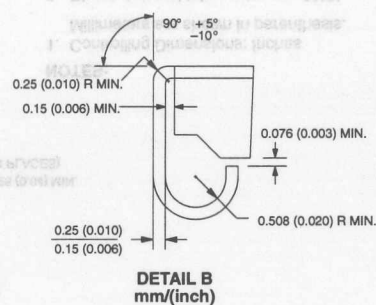
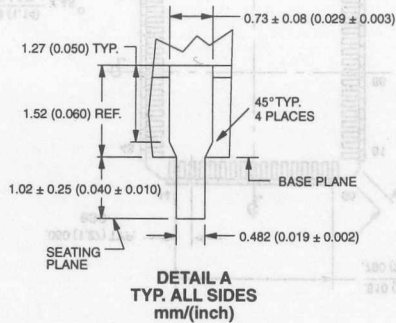
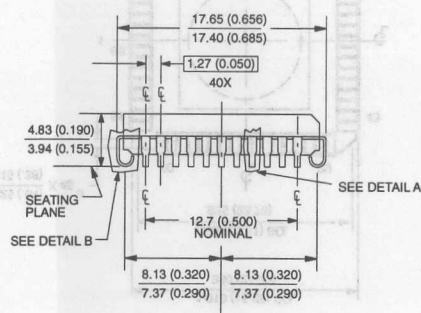
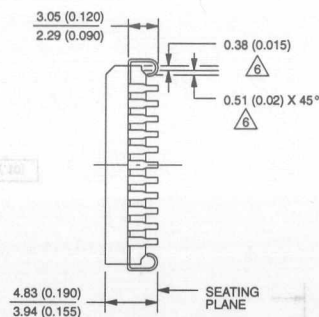
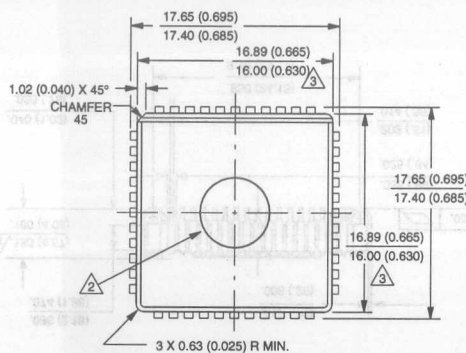
- 3** When a window lid is used, the overall package thickness must increase by a minimum of .010 inch (0.25mm) and a maximum of .040 inch (1.020mm).

## Package outlines

## 1472A 44-PIN CERQUAD J-BEND (K) PACKAGE

## NOTES:

1. All dimensions and tolerances to conform to ANSI Y14.5-1982.
2. UV window is optional.
3. Dimensions do not include glass protrusion. Glass protrusion to be 0.005 inches maximum on each side.
4. Controlling dimension millimeters.
5. All dimensions and tolerances include lead trim offset and lead plating finish.
6. Backside solder relief is optional and dimensions are for reference only.



1990



1. All dimensions and tolerances to conform to ANSI Y14.5—1982.
2. UV window is optional.
3. Dimensions do not include glass protrusion. Glass protrusion to be 0.005 inches maximum on each side.
4. Controlling dimension millimeters.
5. All dimensions and tolerances include lead trim offset and lead plating finish.
6. Backside solder relief is optional and dimensions are for reference only.

2. UV window is optional.

3. Dimensions do not include glass protrusion.  
Glass protrusion to be 0.005 inches maximum  
on each side.

4. Controlling dimension millimeters.

5. All dimensions and tolerances include lead trim offset and lead plating finish.

6. Backside solder relief is optional and dimensions are for reference only.

841387

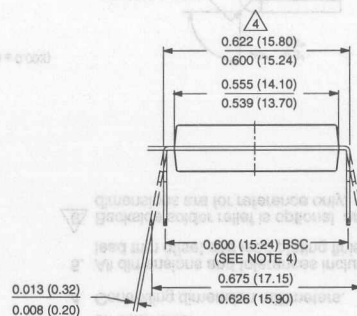
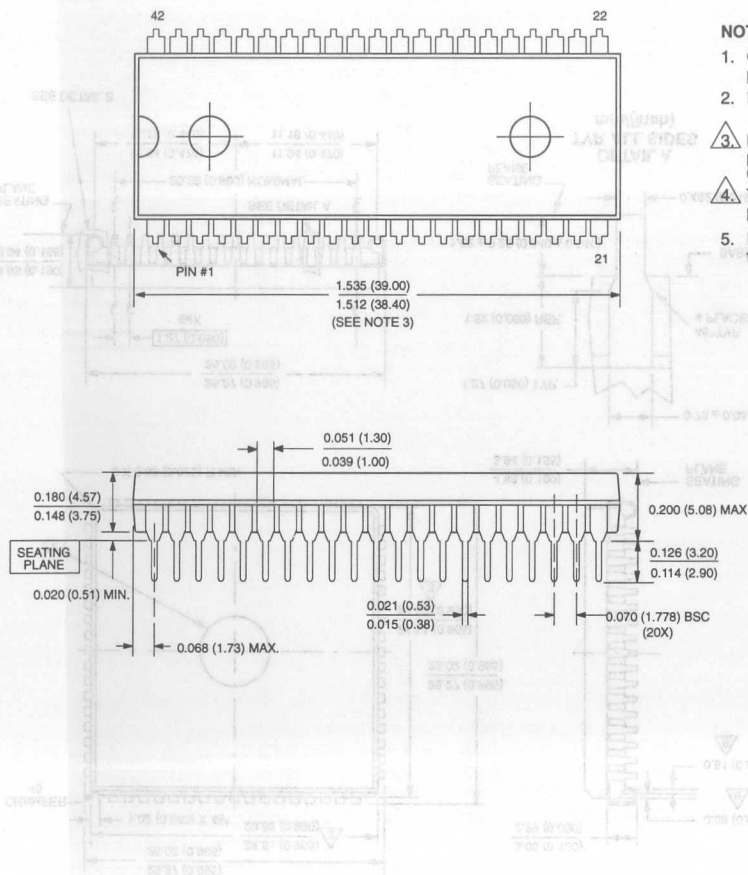
## Package outlines

1680







42-PIN PLASTIC SSOP (SHRINK SMALL OUTLINE PACKAGE) DUAL IN-LINE (D/K) PACKAGE

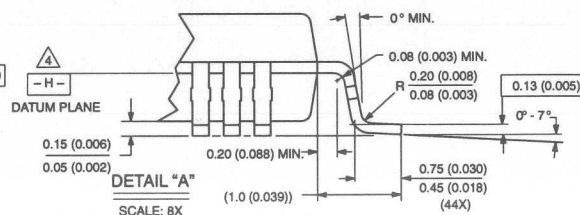
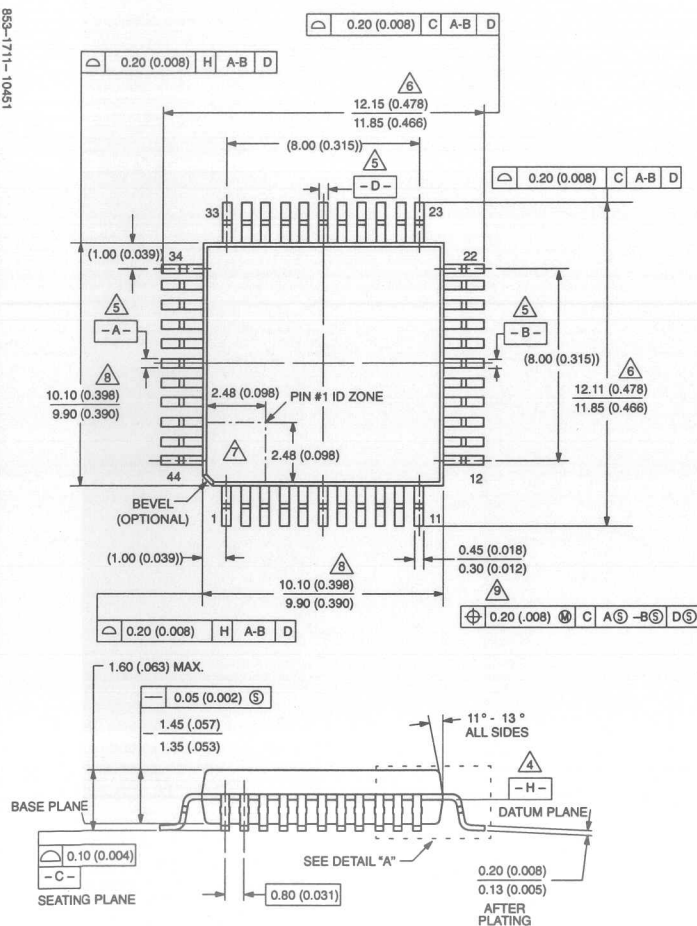
## NOTES

1. Controlling dimension: inches. Metric dimensions are shown in parentheses.
2. Package dimensions conform to Philips envelope SOT-270.
3. Molded body length and width do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.010 inch (0.25mm) per side.
4. These dimensions measured with the leads constrained to be perpendicular to seating plane.
5. Pin numbers start with Pin #1 and continues counterclockwise to Pin #42 when viewed from the top.



**NOTES:**

1. Package dimensions conform to JEDEC registration MO-136-1992.
  2. Controlling dimensions: millimeters. Dimensions in inches are shown in parentheses.
  3. Dimension and tolerancing per ANSI Y14.5M-1982.
-  Datum plane "H" is located at the mold parting line and is coincident with the lead where the lead exits the plastic body at the bottom of the parting line.
-  Datums "A-B" and "D" to be determined at datum plane "H".
-  To be determined at seating plane "C".
-  Details of Pin 1 identifier are optional but must be located within the zone indicated.
-  Package body dimensions do not include mold protrusion. Allowable protrusion is 0.25mm per side, but it includes mold mismatch.
-  Lead width does not include lead protrusion. Allowable dambar protrusion shall be 0.08 mm / 0.003" total in excess of this dimension at maximum material condition. Dambar cannot be located on the lower radius or the foot.







Philips Semiconductors

# **Section 10**

## **North American Sales Offices, Representatives and Distributors**

**80C51-Based  
8-Bit Microcontrollers**



## North American Sales Offices, Representatives and Distributors

**PHILIPS  
SEMICONDUCTORS**  
811 East Arques Avenue  
P.O. Box 3409  
Sunnyvale, CA 94088-3409

**ALABAMA**  
**Huntsville**  
Philips Semiconductors  
Phone: (205) 464-0111

Elcom, Inc.  
Phone: (205) 830-4001

**ARIZONA**  
**Scottsdale**  
Thom Luke Sales, Inc.  
Phone: (602) 451-5400

**CALIFORNIA**  
**Calabasas**  
Philips Semiconductors  
Phone: (818) 880-6304

**Irvine**  
Philips Semiconductors  
Phone: (714) 453-0770

**Loomis**  
B.A.E. Sales, Inc.  
Phone: (916) 652-6777

**San Diego**  
Philips Semiconductors  
Phone: (619) 560-0242

**San Jose**  
B.A.E. Sales, Inc.  
Phone: (408) 452-8133

**Sunnyvale**  
Philips Semiconductors  
Phone: (408) 991-3737

**COLORADO**  
**Englewood**  
Philips Semiconductors  
Phone: (303) 792-9011

Thom Luke Sales, Inc.  
Phone: (303) 649-9717

**CONNECTICUT**  
**Wallingford**  
JEBCO  
Phone: (203) 265-1318

**FLORIDA**  
**Oviedo**  
Conley and Assoc., Inc.  
Phone: (407) 365-3283

**GEORGIA**  
**Atlanta**  
Philips Semiconductors  
Phone: (404) 594-1392

**Norcross**  
Elcom, Inc.  
Phone: (404) 447-8200

**ILLINOIS**  
**Hoffman Estates**  
Micro-Tex, Inc.  
Phone: (708) 765-3000

**Itasca**  
Philips Semiconductors  
Phone: (708) 250-0050

**INDIANA**  
**Indianapolis**  
Mohrfield Marketing, Inc.  
Phone: (317) 546-6969

**Kokomo**  
Philips Semiconductors  
Phone: (317) 459-5355

**MARYLAND**  
**Columbia**  
Third Wave Solutions, Inc.  
Phone: (410) 290-5990

**MASSACHUSETTS**  
**Chelmsford**  
JEBCO  
Phone: (508) 256-5800

**Westford**  
Philips Semiconductors  
Phone: (508) 692-6211

**MICHIGAN**  
**Monroe**  
S-J Associates  
Phone: (313) 242-0450

**Novi**  
Philips Semiconductors  
Phone: (313) 347-1700

**MINNESOTA**  
**Bloomington**  
High Technology Sales  
Phone: (612) 844-9933

**MISSOURI**  
**Bridgeton**  
Centech, Inc.  
Phone: (314) 291-4230

**Raytown**  
Centech, Inc.  
Phone: (816) 358-8100

**NEW JERSEY**  
**Toms River**  
Philips Semiconductors  
Phone: (908) 505-1200

**NEW YORK**  
**Ithaca**  
Bob Dean, Inc.  
Phone: (607) 257-1111

**Rockville Centre**  
S-J Associates  
Phone: (516) 536-4242

**Wappingers Falls**  
Philips Semiconductors  
Phone: (914) 297-4074

Bob Dean, Inc.  
Phone: (914) 297-6406

**NORTH CAROLINA**  
**Greensboro**  
ADI, Inc.  
Phone: (919) 273-8887

**Matthews**  
ADI, Inc.  
Phone: (704) 847-4323

**Smithfield**  
ADI, Inc.  
Phone: (919) 934-8136

**OHIO**  
**Aurora**  
S-J Associates, Inc.  
Phone: (216) 562-2050

**Columbus**  
S-J Associates, Inc.  
Phone: (614) 885-6700

**Kettering**  
S-J Associates, Inc.  
Phone: (513) 298-7322

**Parma**  
S-J Associates, Inc.  
Phone: (216) 888-7004

**Toledo**  
S-J Associates, Inc.  
Phone: (313) 242-0450

**West Carrollton**  
S-J Associates, Inc.  
Phone: (513) 438-1700

**OREGON**  
**Beaverton**  
Philips Semiconductors  
Phone: (503) 627-0110

Western Technical Sales  
Phone: (503) 644-8860

**PENNSYLVANIA**  
**Erie**  
S-J Associates, Inc.  
Phone: (216) 888-7004

**Hatboro**  
Delta Technical Sales, Inc.  
Phone: (215) 957-0600

**Pittsburgh**  
S-J Associates, Inc.  
Phone: (216) 888-7004

**Plymouth Meeting**  
Philips Semiconductors  
Phone: (215) 825-4404

**TENNESSEE**  
**Dandridge**  
Philips Semiconductors  
Phone: (615) 397-5053

**TEXAS**  
**Austin**  
Philips Semiconductors  
Phone: (512) 339-9945

**Austin**  
Synergistic Sales, Inc.  
Phone: (512) 346-2122

**Houston**  
Synergistic Sales, Inc.  
Phone: (713) 937-1990

**Richardson**  
Philips Semiconductors  
Phone: (214) 644-1610

**Richardson**  
Synergistic Sales, Inc.  
Phone: (214) 644-3500

**UTAH**  
**Salt Lake City**  
Electrodyn  
Phone: (801) 264-8050

**WASHINGTON**  
**Bellevue**  
Western Technical Sales  
Phone: (206) 641-3900

**Spokane**  
Western Technical Sales  
Phone: (509) 922-7600

**WISCONSIN**  
**Waukesha**  
Micro-Tex, Inc.  
Phone: (414) 542-5352

**CANADA**  
**PHILIPS  
SEMICONDUCTORS  
CANADA, LTD.**

**Calgary, Alberta**  
Tech-Trek, Ltd.  
Phone: (403) 241-1719

**Kanata, Ontario**  
Philips Semiconductors  
Phone: (613) 599-8720

**Mississauga, Ontario**  
Tech-Trek, Ltd.  
Phone: (416) 238-0366

**Richmond, B.C.**  
Tech-Trek, Ltd.  
Phone: (604) 276-8735

**Ville St. Laurent, Quebec**  
Tech-Trek, Ltd.  
Phone: (514) 337-7540

**MEXICO**  
**Anzures Section**  
Philips Components  
Phone: 52-5-533-3858

**El Paso, TX**  
Philips Components  
Phone: (915) 775-4020

**PUERTO RICO**  
**Santurce**  
Mectron Group  
Phone: (809) 723-6165

### DISTRIBUTORS

**Contact one of our  
local distributors:**

Almac/Arrow Electronics

Anthem Electronics

Arrow/Schweber Electronics

Future Electronics

Gerber Electronics

Hamilton Hallmark

Marshall Industries

Wyle Laboratories, EMG

Zeus Electronics

## Data handbook system

## Appendix A

## DATA HANDBOOK SYSTEM

Philips Semiconductors data handbooks contain all pertinent data available at the time of publication and each is revised and reissued regularly.

Loose data sheets are sent to subscribers to keep them up-to-date on additions or alterations made during the lifetime of a data handbook.

Catalogs are available for selected product ranges (some catalogs are also on floppy discs).

Our data handbook titles are listed here.

## Integrated Circuits

Book	Title
IC01	Semiconductors for Radio and Audio Systems
IC02	Semiconductors for Television and Video Systems
IC03	Semiconductors for Telecom Systems
IC04	CMOS HE4000B Logic Family
IC05	Advanced Low-power Schottky (ALS) Logic Series
IC06	High-speed CMOS Logic Family
IC08	100K ECL Logic Families
IC10	Memories
IC11	General-purpose/Linear ICs
IC12	Display Drivers and Microcontroller Peripherals (planned)
IC13	Programmable Logic Devices (PLD)
IC14	8048-based 8-bit Microcontrollers
IC15	FAST TTL Logic Series
IC16	ICs for Clocks and Watches
IC18	Semiconductors for In-car Electronics and General Industrial Applications (planned)
IC17	RF/Wireless Communications
IC19	Semiconductors for Datacom: LANs, UARTs, Multi-protocol Controllers and Fibre Optics
IC20	80C51-based 8-bit Microcontrollers
IC21	68000-based 16-bit Microcontrollers (planned)
IC22	ICs for Multi-media Systems (planned)
IC23	QUBiC Advanced BiCMOS Bus Interface Logic ABT, MULTIBYTE™
IC24	Low Voltage CMOS & BiCMOS Logic

## Discrete Semiconductors

Book	Title
SC01	Diodes
SC02	Power Diodes
SC03	Thyristors and Triacs
SC04	Small-signal Transistors
SC05	Low-frequency Power Transistors and Hybrid IC Power Modules
SC06	High-voltage and Switching NPN Power Transistors
SC07	Small-signal Field-effect Transistors
SC08a	RF Power Bipolar Transistors
SC08b	RF Power MOS Transistors
SC09	RF Power Modules
SC10	Surface Mounted Semiconductors
SC13	Power MOS Transistors including TOFETs and IGBTs
SC14	RF Wideband Transistors, Video Transistors and Modules
SC15	Microwave Transistors
SC16	Wideband Hybrid IC Modules
SC17	Semiconductor Sensors

## Professional Components

PC01	High-power Klystrons and Accessories
PC06	Circulators and Isolators

## MORE INFORMATION FROM PHILIPS SEMICONDUCTORS?

For more information about Philips Semiconductors data handbooks, catalogs and subscriptions contact your nearest Philips Semiconductors national organization, select from the **address list on the back cover of this handbook**. Product specialists are at your service and inquiries are answered promptly.



## Data handbook system

## Appendix A

## OVERVIEW OF PHILIPS COMPONENTS DATA HANDBOOKS

Our sister product division, Philips Components, also has a comprehensive data handbook system to support their products. Their data handbook titles are listed here.

### Display Components

Book	Title
DC01	Colour Display Components Colour TV Picture Tubes and Assemblies Colour Monitor Tube Assemblies
DC02	Monochrome Monitor Tubes and Deflection Units
DC03	Television Tuners, Coaxial Aerial Input Assemblies
DC05	Flyback Transformers, Mains Transformers and General-purpose FXC Assemblies

### Magnetic Products

MA01	Soft Ferrites
MA03	Piezoelectric Ceramics Specialty Ferrites
MA04	Dry-reed Switches

### Passive Components

PA01	Electrolytic Capacitors
PA02	Varistors, Thermistors and Sensors
PA03	Potentiometers and Switches
PA04	Variable Capacitors
PA05	Film Capacitors
PA06	Ceramic Capacitors
PA07	Quartz Crystals for Special and Industrial Applications
PA08	Fixed Resistors
PA10	Quartz Crystals for Automotive and Standard Applications
PA11	Quartz Oscillators

### Professional Components

PC04	Photo Multipliers
PC05	Plumbicon Camera Tubes and Accessories
PC07	Vidicon and Newvicon Camera Tubes and Deflection Units
PC08	Image Intensifiers
PC12	Electron Multipliers

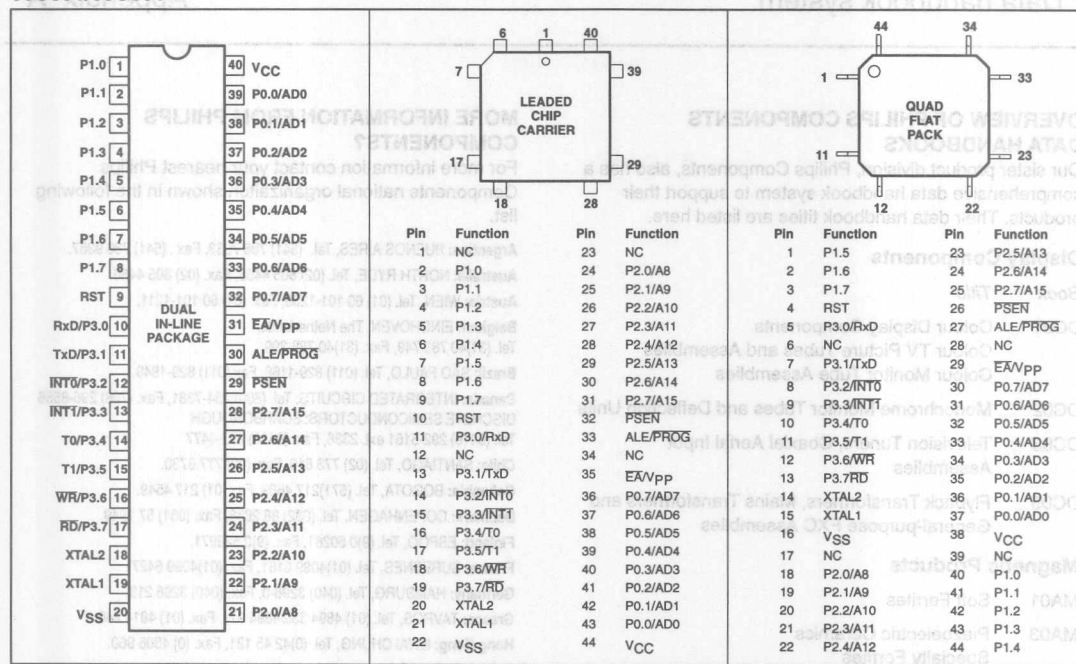
## MORE INFORMATION FROM PHILIPS COMPONENTS?

For more information contact your nearest Philips Components national organization shown in the following list.

**Argentina:** BUENOS AIRES, Tel. (541) 786 7633, Fax. (541) 786 9367.  
**Australia:** NORTH RYDE, Tel. (02) 805 4455, Fax. (02) 805 4466.  
**Austria:** WIEN, Tel. (01) 60 101-1236, Fax. (01) 60 101-1211.  
**Belgium:** EINDHOVEN, The Netherlands, Tel. (31)40 783 749, Fax. (31)40 788 399.  
**Brazil:** SÃO PAULO, Tel. (011) 829-1166, Fax (011) 829-1849.  
**Canada:** INTEGRATED CIRCUITS: Tel. (800) 234-7381, Fax. (708) 296-8556  
 DISCRETE SEMICONDUCTORS: SCARBOROUGH  
 Tel. (0416) 292-5161 ext. 2336, Fax. (0416) 292-4477  
**Chile:** SANTIAGO, Tel. (02) 773 816, Fax. (02) 777 6730.  
**Columbia:** BOGOTÁ, Tel. (571)217 4609, Fax (01) 217 4549.  
**Denmark:** COPENHAGEN, Tel. (032) 88 2636, Fax. (031) 57 1949.  
**Finland:** ESPOO, Tel. (9)0 50261, Fax. (9)0 520971.  
**France:** SURESNES, Tel. (01)4099 6161, Fax, (01)4099 6427.  
**Germany:** HAMBURG, Tel. (040) 3296-0, Fax. (040) 3296 213.  
**Greece:** TAVROS, Tel. (01) 4894 339/4894 911, Fax. (01) 4814 240.  
**Hong Kong:** KWAI CHUNG, Tel. (0)42 45 121, Fax. (0) 4806 960.  
**India:** BOMBAY, Tel. (022) 4938 541, Fax. (022) 4938 722.  
**Indonesia:** JAKARTA, Tel. (021) 5201 122, Fax. (021) 5205 189.  
**Ireland:** DUBLIN, Tel. (01) 640 000, Fax. (01) 640 200.  
**Italy:** MILANO, Tel. (02) 6752.1, Fax. (02) 6752.3350.  
**Japan:** TOKYO, Tel. (03) 3740 5101, Fax. (03) 3740 0570.  
**Korea (Republic of):** SEOUL, SEOUL, Tel. (02) 794-5011, Fax. (02) 798-8022.  
**Malaysia:** SELANGOR, Tel. (03) 757 5511, Fax. (03) 757 4880.  
**Mexico:** EL PASO, TX 79905, Tel. 9-5(800) 234-7381, Fax. (708) 296-8556  
**Netherlands:** EINDHOVEN, Tel. (040) 78 37 49, Fax. (040)78 83 99.  
**New Zealand:** AUKLAND, Tel. (09) 849-4160, Fax. (09) 849-7811.  
**Norway:** OSLO, Tel. (22) 74 8000, Fax (22)74 8341.  
**Pakistan:** KARACHI, Tel. (021) 577 039, Fax. (021) 569 1832.  
**Philippines:** MANILA, Tel. (02) 810 0161, Fax. (02) 817 3474.  
**Portugal:** LISBOA, Tel. (01) 683 121, Fax. (01) 658 013.  
**Singapore:** SINGAPORE, Tel. (65) 350 2000, Fax. (65) 251 6500.  
**South Africa:** JOHANNESBURG, Tel. (011) 470-5433, Fax. (011) 470-5494.  
**Spain:** BARCELONA, Tel. (03) 301 6312, Fax. (03) 301 4243.  
**Sweden:** STOCKHOLM, Tel. (08)632 2000, Fax. (08)632 2745.  
**Switzerland:** ZÜRICH, Tel. (01) 488 2211, Fax. (01) 481 7730.  
**Taiwan:** TAIPEI, Tel. (2) 388 7666, Fax. (2)382 4382.  
**Thailand:** BANGKOK, Tel. (2)399-3280 to 9, (2)398-2083, Fax. (2)398-2080.  
**Turkey:** ISTANBUL, Tel. (0212) 279 2770, Fax. (0212) 269 3094.  
**United Kingdom:** LONDON, Tel. (071) 436 4144, Fax. (071) 323 0342.  
**United States:** INTEGRATED CIRCUITS: SUNNYVALE, CA  
 Tel. (800) 234-7381, Fax. (708) 296-8556  
 DISCRETE SEMICONDUCTORS: RIVIERA BEACH, FL  
 Tel. (800) 447-3762 and (407)881-3200, Fax. (407) 881-3300  
**Uruguay:** MONTEVIDEO, Tel. (02) 70-4044, Fax (02) 92 0601.  
 For all other countries apply to: **Philips Components.**  
 Marketing Communications, Building BAF-1,  
 P.O. Box 218, 5600 MD, EINDHOVEN, The Netherlands  
 Telex 35000 phnlcl, Fax. +31-40-724825.

## Appendix B

8031AH/8051AH, 80C31/80C51/87C51, 83C504/87C504, 80C652/83C652/87C652\*, 83C654/87C654\*, 80C851/83C851

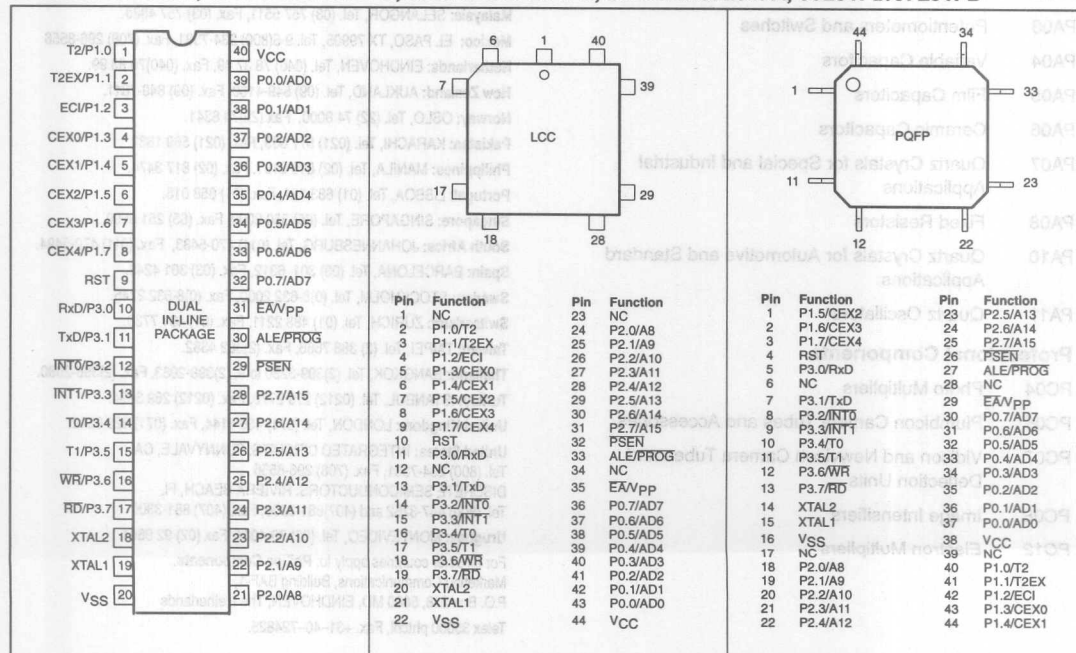


\* P1.6 and P1.7 have the alternate functions SCL and SDA, respectively, on the 80C652/83C652/87C652 and 83C654/87C654.

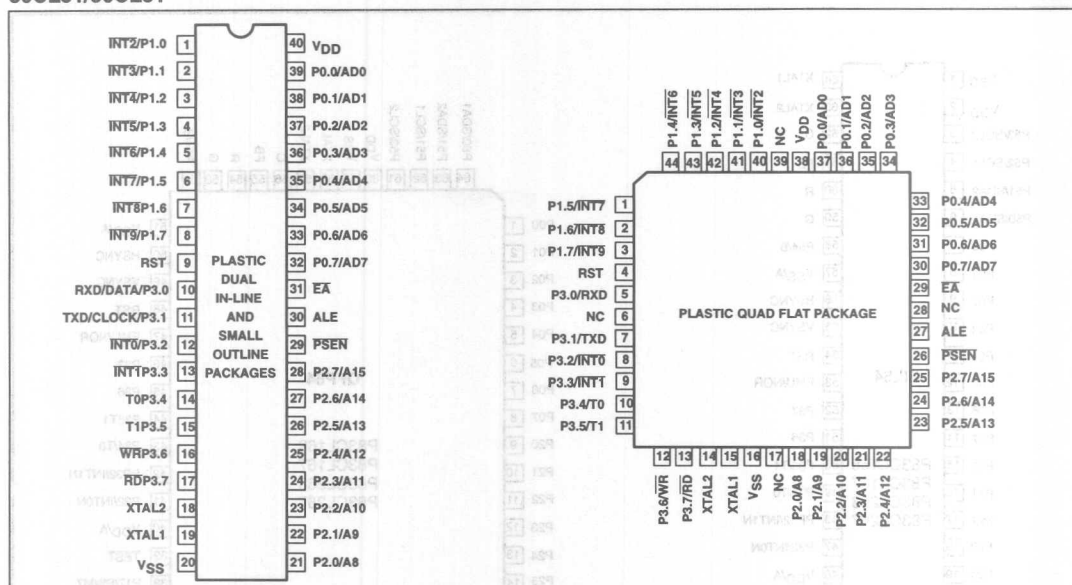
### NOTES TO QFP ONLY:

- Due to EMC improvements, it is advised to connect pins 6, 16, 28, 39 to VSS on the 80C652/83C652, and 83C654.

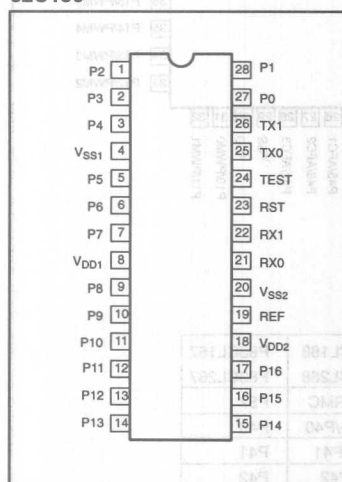
83C51FA/87C51FA, 83C51FB/87C51FB, 83C51FC/87C51FC, 83L51FA/87L51FA, 83L51FB/87L51FB



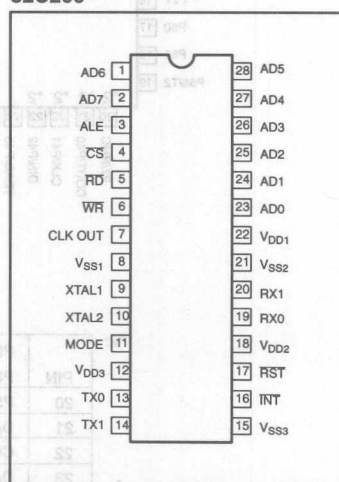
## 80CL31/80CL51



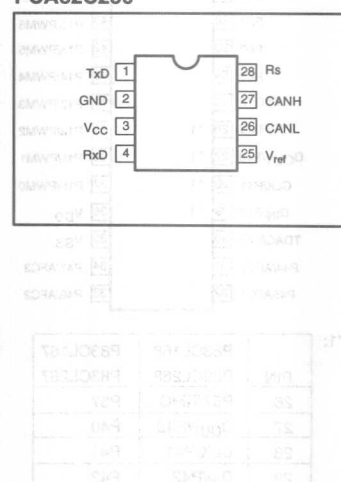
## 82C150



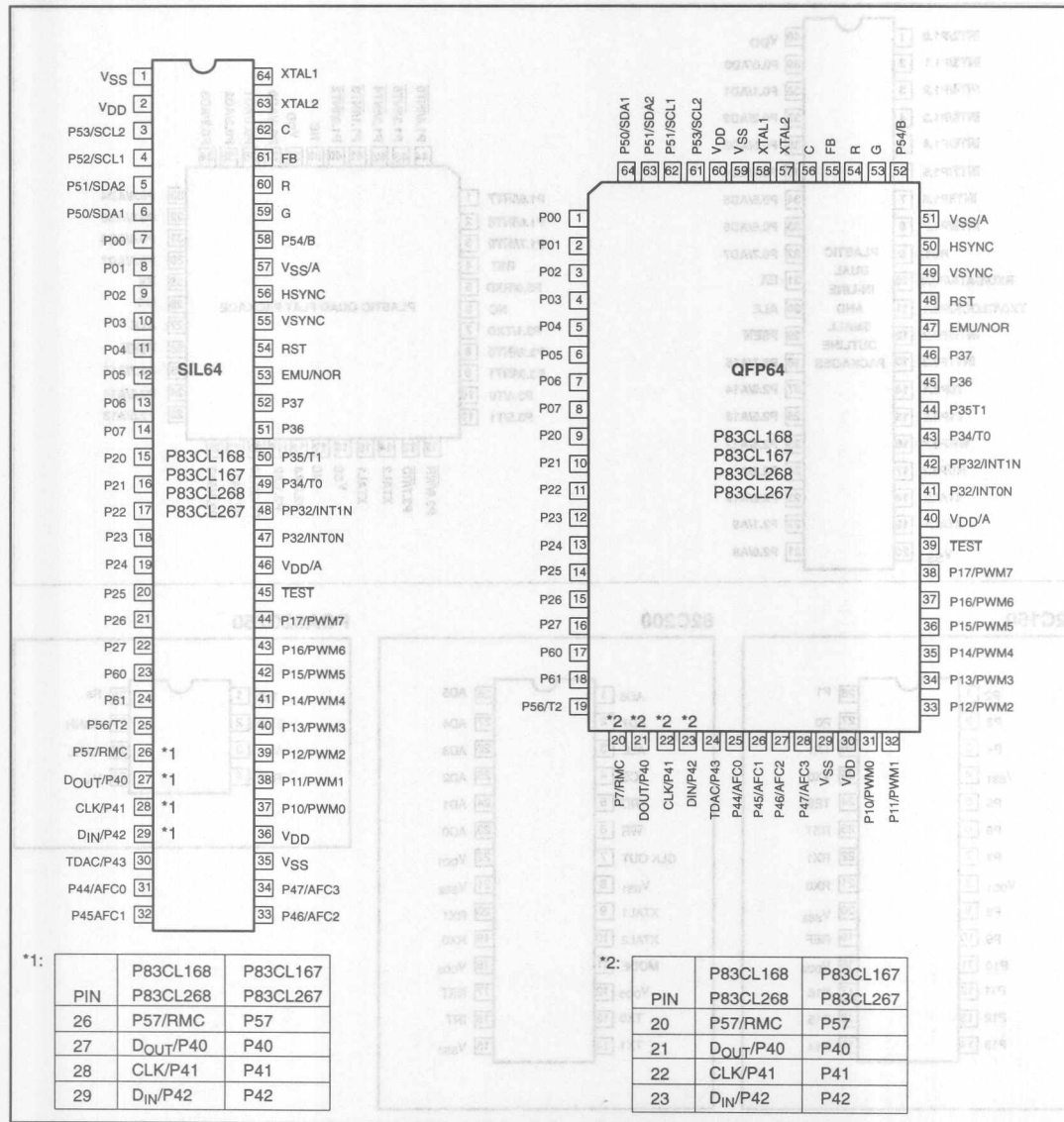
## 82C200



## PCA82C250

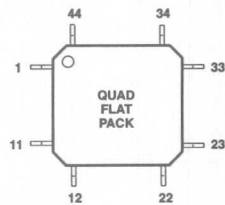


P83CL168/167/268/267

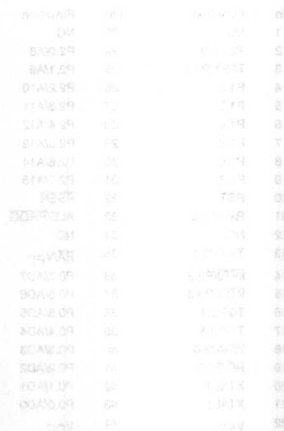
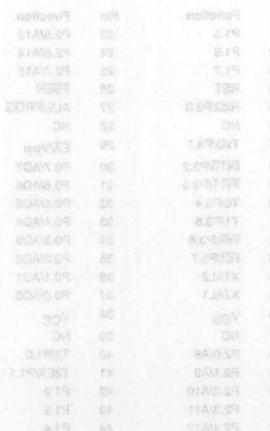
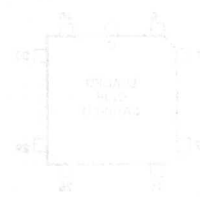
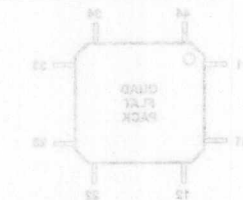


80CE654/83CE654

80CE654/83CE654, 80CE654/83CE654, 80CE654/83CE654



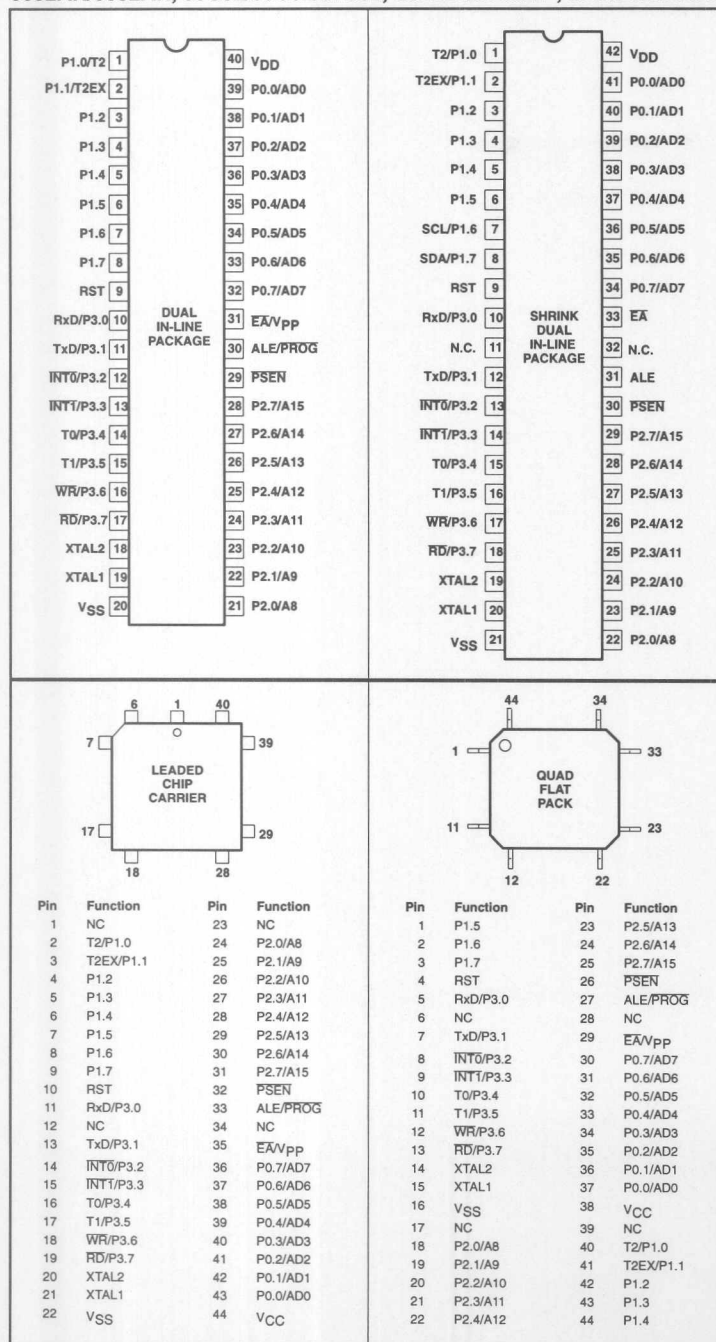
Pin	Function	Pin	Function
1	P1.5	23	P2.5/A13
2	P1.6/SCL	24	P2.6/A14
3	P1.7/SDA	25	P2.7/A15
4	RST	26	PSEN
5	P3.0/RxD	27	ALE
6	VSS4	28	VSS2
7	P3.1/TxD	29	EA
8	P3.2/INT0	30	P0.7/AD7
9	P3.3/INTT	31	P0.6/AD6
10	P3.4/T0	32	P0.5/AD5
11	P3.5/T1	33	P0.4/AD4
12	P3.6/WR	34	P0.3/AD3
13	P3.7/RD	35	P0.2/AD2
14	XTAL2	36	P0.1/AD1
15	XTAL1	37	P0.0/AD0
16	VSS1	38	VDD2
17	VDD1	39	VSS3
18	P2.0/A8	40	P1.0
19	P2.1/A9	41	P1.1
20	P2.2/A10	42	P1.2
21	P2.3/A11	43	P1.3
22	P2.4/A12	44	P1.4



P1.5 and P1.7 have the alternate functions SCL and SDA, respectively, on the 80CE654/83CE654 and 80CE654/83CE654.



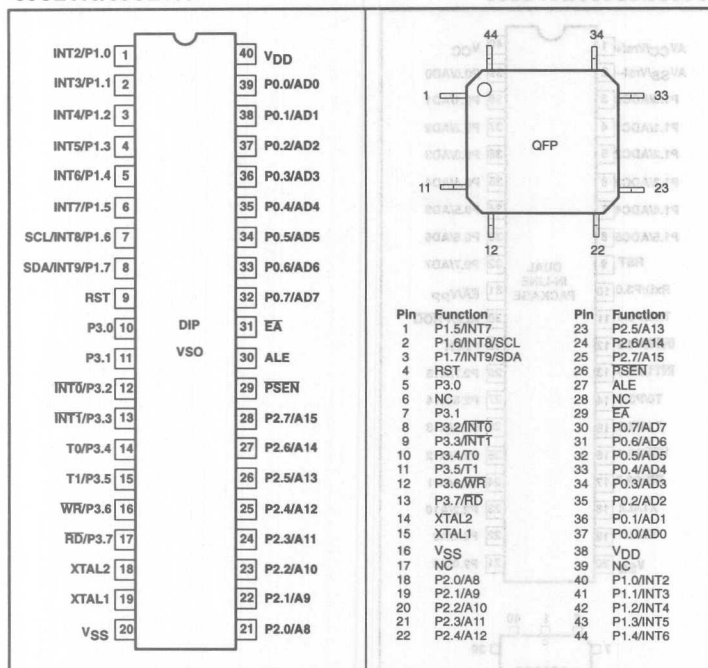
8032AH/8052AH, 80C32/80C52/87C52, 83C524/87C524\*, 80C528/83C528/87C528\*



\* P1.6 and P1.7 have the alternate functions SCL and SDA, respectively, on the 83C524/87C524 and 80C528/83C528/87C528.

# Appendix B

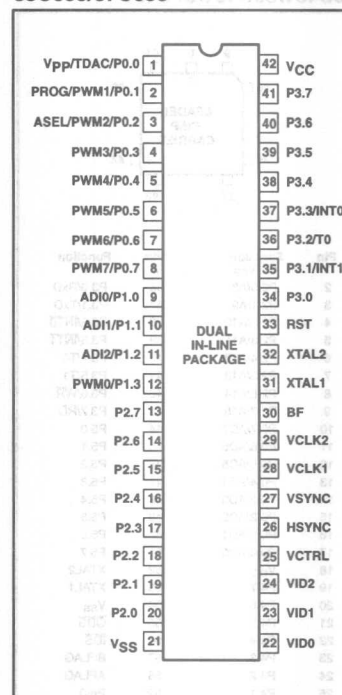
## 80CL410/83CL410\*



\* P1.6 and P1.7 have the alternate functions SCL and SDA, respectively, on the 80CL410/83CL410.

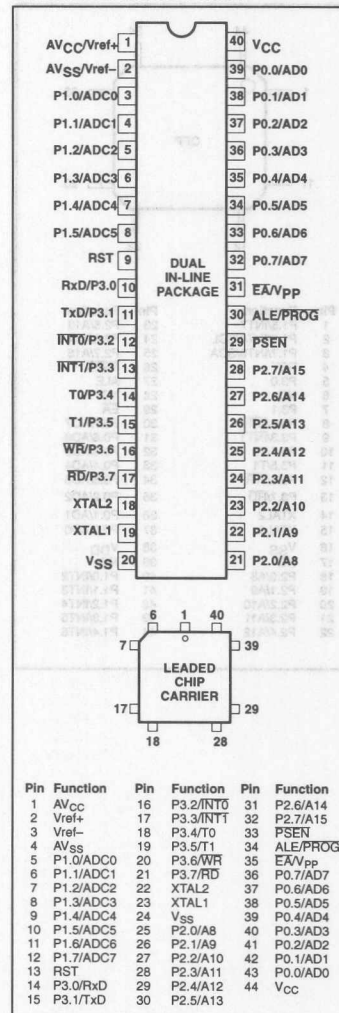
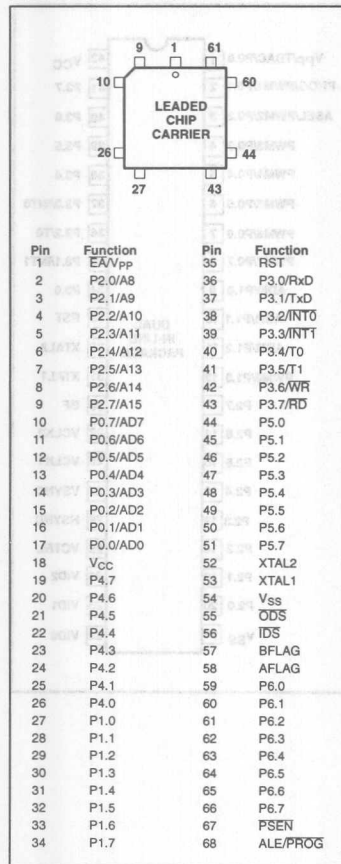
# Appendix B

## 83C055/87C055

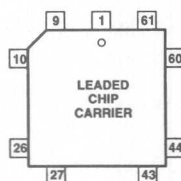


80C451/83C451/87C451

80C550/83C550/87C550



## 80C552/83C552/87C552\*, 80C562/83C562



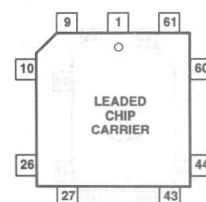
Pin	Function	Pin	Function
1	P5.0/ADC0	35	XTAL1
2	VDD	36	VSS
3	STADC	37	VSS
4	PWM0	38	NC
5	PWM1	39	P2.0/A08
6	EW	40	P2.1/A09
7	P4.0/CMSR0	41	P2.2/A10
8	P4.1/CMSR1	42	P2.3/A11
9	P4.2/CMSR2	43	P2.4/A12
10	P4.3/CMSR3	44	P2.5/A13
11	P4.4/CMSR4	45	P2.6/A14
12	P4.5/CMSR5	46	P2.7/A15
13	P4.6/CMT0	47	PSEN
14	P4.7/CMT1	48	ALE/PROG
15	RST	49	EA/Vpp
16	P1.0/CT0I	50	P0.7/AD7
17	P1.1/CT1I	51	P0.6/AD6
18	P1.2/CT2I	52	P0.5/AD5
19	P1.3/CT3I	53	P0.4/AD4
20	P1.4/T2	54	P0.3/AD3
21	P1.5/RT2	55	P0.2/AD2
22	P1.6/SCL	56	P0.1/AD1
23	P1.7/SDA	57	P0.0/AD0
24	P3.0/RxD	58	AVref-
25	P3.1/TxD	59	AVref+
26	P3.2/INT0	60	AVSS
27	P3.3/INT1	61	AVDD
28	P3.4/T0	62	P5.7/ADC7
29	P3.5/T1	63	P5.6/ADC6
30	P3.6/WR	64	P5.5/ADC5
31	P3.7/RD	65	P5.4/ADC4
32	NC	66	P5.3/ADC3
33	NC	67	P5.2/ADC2
34	XTAL2	68	P5.1/ADC1



Pin	Function	Pin	Function
1	P4.1/CMSR1	41	P2.3/A11
2	P4.2/CMSR2	42	P2.4/A12
3	NC	43	NC
4	P4.3/CMSR3	44	NC
5	P4.4/CMSR4	45	P2.5/A13
6	P4.5/CMSR5	46	P2.6/A14
7	P4.6/CMT0	47	P2.7/A15
8	P4.7/CMT1	48	PSEN
9	RST	49	ALE/PROG
10	P1.0/CT0I	50	EA/Vpp
11	P1.1/CT1I	51	P0.7/AD7
12	P1.2/CT2I	52	P0.6/AD6
13	P1.3/CT3I	53	P0.5/AD5
14	P1.4/T2	54	P0.4/AD4
15	P1.5/RT2	55	P0.3/AD3
16	P1.6/SCL	56	P0.2/AD2
17	P1.7/SDA	57	P0.1/AD1
18	P3.0/RxD	58	P0.0/AD0
19	P3.1/TxD	59	AVref-
20	P3.2/INT0	60	AVref+
21	NC	61	AVSS
22	NC	62	NC
23	P3.3/INT1	63	AVDD
24	P3.4/T0	64	P5.7/ADC7
25	P3.5/T1	65	P5.6/ADC6
26	P3.6/WR	66	P5.5/ADC5
27	P3.7/RD	67	P5.4/ADC4
28	NC	68	P5.3/ADC3
29	NC	69	P5.2/ADC2
30	NC	70	P5.1/ADC1
31	XTAL2	71	P5.0/ADC0
32	XTAL1	72	VDD
33	IC	73	IC
34	VSS	74	STADC
35	VSS	75	PWM0
36	VSS	76	PWM1
37	NC	77	EW
38	P2.0/A08	78	NC
39	P2.1/A09	79	NC
40	P2.2/A10	80	P4.0/CMSR0

NC = not connected  
IC = internally connected (do not use)

## 80C592/83C592/87C592



Pin	Function	Pin	Function
1	P5.0/ADC0	35	VSS
2	VDD	36	P2.0/A08
3	STADC	37	P2.1/A09
4	PWM0	38	P2.2/A10
5	PWM1	39	P2.3/A11
6	EW	40	P2.4/A12
7	P4.0/CMSR0	41	P2.5/A13
8	P4.1/CMSR1	42	P2.6/A14
9	P4.2/CMSR2	43	P2.7/A15
10	P4.3/CMSR3	44	PSEN
11	P4.4/CMSR4	45	ALE/PROG
12	P4.5/CMSR5	46	EA/Vpp
13	P4.6/CMT0	47	P0.7/AD7
14	P4.7/CMT1	48	P0.6/AD6
15	RST	49	P0.5/AD5
16	P1.0/CT0I/INT2	50	P0.4/AD4
17	P1.1/CT1I/INT3	51	P0.3/AD3
18	P1.2/CT2I/INT4	52	P0.2/AD2
19	P1.3/CT3I/INT5	53	P0.1/AD1
20	P1.4/T2	54	P0.0/AD0
21	P1.5/RT2	55	REF
22	CVSS	56	CRX1
23	P1.6/CTX0	57	CRX0
24	P1.7/CTX1	58	AVref-
25	P3.0/RxD	59	AVref+
26	P3.1/TxD	60	AVSS
27	P3.2/INT0	61	AVDD
28	P3.3/INT1	62	P5.7/ADC7
29	P3.4/T0	63	P5.6/ADC6
30	P3.5/T1	64	P5.5/ADC5
31	P3.6/WR	65	P5.4/ADC4
32	P3.7/RD	66	P5.3/ADC3
33	XTAL2	67	P5.2/ADC2
34	XTAL1	68	P5.1/ADC1

\* P1.6 and P1.7 have the alternate functions SCL and SDA, respectively, on the 80C552/83C552/87C552.



Pin	Function	Pin	Function
1	AVref-	41	P3.0/RXD/DATA
2	AVref+	42	P3.1/TXD/CLOCK
3	AVss1	43	P3.2/INT0
4	AVdd1	44	P3.3/INT1
5	P5.7/ADC7	45	P3.4/T0
6	P5.6/ADC6	46	P3.5/T1
7	P5.5/ADC5	47	P3.6/WR
8	P5.4/ADC4	48	P3.7/RD
9	P5.3/ADC3	49	NC
10	P5.2/ADC2	50	NC
11	P5.1/ADC1	51	XTAL2
12	P5.0/ADC0	52	XTAL1
13	Vss1	53	Vdd3
14	Vdd1	54	Vss3
15	ADEXS	55	P2.0/A8
16	PWM0	56	P2.1/A9
17	PWMT	57	P2.2/A10
18	EW	58	P2.3/A11
19	P4.0/CMSR0	59	P2.4/A12
20	P4.1/CMSR1	60	P2.5/A13
21	P4.2/CMSR2	61	P2.6/A14
22	P4.3/CMSR3	62	P2.7/A15
23	RSTOUT	63	PSEN
24	P4.4/CMSR4	64	ALE/WE
25	P4.5/CMSR5	65	EA
26	P4.6/CMT0	66	Vdd4
27	P4.7/CMT1	67	Vss4
28	Vdd2	68	P0.7/AD7
29	Vss2	69	P0.6/AD6
30	RSTIN	70	P0.5/AD5
31	P1.0/CT0/INT2	71	P0.4/AD4
32	P1.1/CT1/INT3	72	P0.3/AD3
33	P1.2/CT2/INT4	73	P0.2/AD2
34	P1.3/CT3/INT5	74	P0.1/AD1
35	P1.4/T2	75	P0.0/AD0
36	P1.5/RT2	76	AVdd2
37	P1.6	77	AVss2
38	P1.7	78	XTAL3
39	SCL	79	XTAL4
40	SDA	80	SELXTAL1

\* only 89CE558 with alternative function



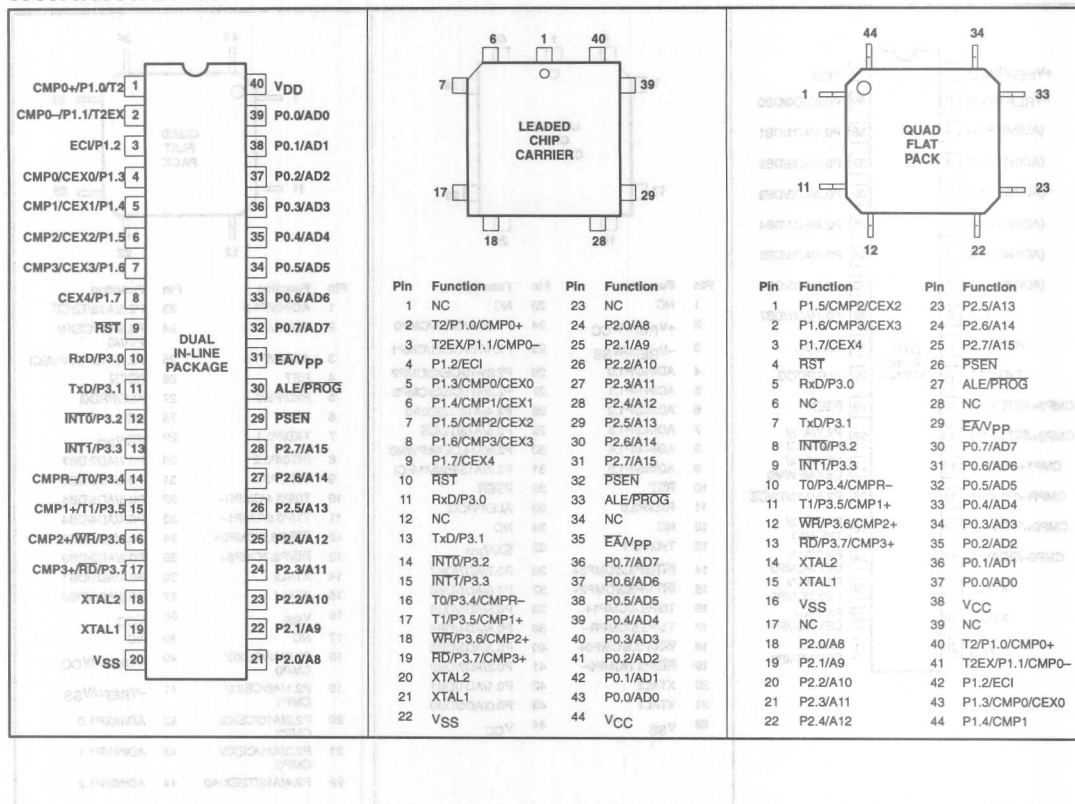
Pin	Function	Pin	Function
1	XTAL1	41	P3.0/RXD/DATA
2	Vdd	42	P3.1/TXD/CLOCK
3	Vss	43	P3.2/INT0
4	NC	44	P3.3/INT1
5	P5.7/ADC7	45	P3.4/T0
6	P5.6/ADC6	46	P3.5/T1
7	P5.5/ADC5	47	P3.6/WR
8	P5.4/ADC4	48	P3.7/RD
9	P5.3/ADC3	49	NC
10	P5.2/ADC2	50	NC
11	P5.1/ADC1	51	XTAL2
12	P5.0/ADC0	52	XTAL1
13	Vss1	53	Vdd3
14	Vdd1	54	Vss3
15	ADEXS	55	P2.0/A8
16	PWM0	56	P2.1/A9
17	PWMT	57	P2.2/A10
18	EW	58	P2.3/A11
19	P4.0/CMSR0	59	P2.4/A12
20	P4.1/CMSR1	60	P2.5/A13
21	P4.2/CMSR2	61	P2.6/A14
22	P4.3/CMSR3	62	P2.7/A15
23	RSTOUT	63	PSEN
24	P4.4/CMSR4	64	ALE/WE
25	P4.5/CMSR5	65	EA
26	P4.6/CMT0	66	Vdd4
27	P4.7/CMT1	67	Vss4
28	Vdd2	68	P0.7/AD7
29	Vss2	69	P0.6/AD6
30	RSTIN	70	P0.5/AD5
31	P1.0/CT0/INT2	71	P0.4/AD4
32	P1.1/CT1/INT3	72	P0.3/AD3
33	P1.2/CT2/INT4	73	P0.2/AD2
34	P1.3/CT3/INT5	74	P0.1/AD1
35	P1.4/T2	75	P0.0/AD0
36	P1.5/RT2	76	AVdd2
37	P1.6	77	AVss2
38	P1.7	78	XTAL3
39	SCL	79	XTAL4
40	SDA	80	SELXTAL1

NC = not connected  
NC = not connected (pin not used)

P1.6 and P1.7 have the alternate functions SCL and SDA, respectively.  
no the 80CE558/83CE558/89CE558



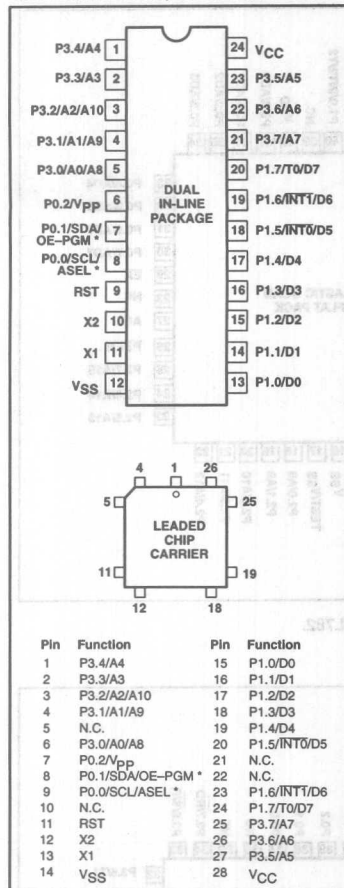
## 80C575/83C575/87C575



83C576/87C576

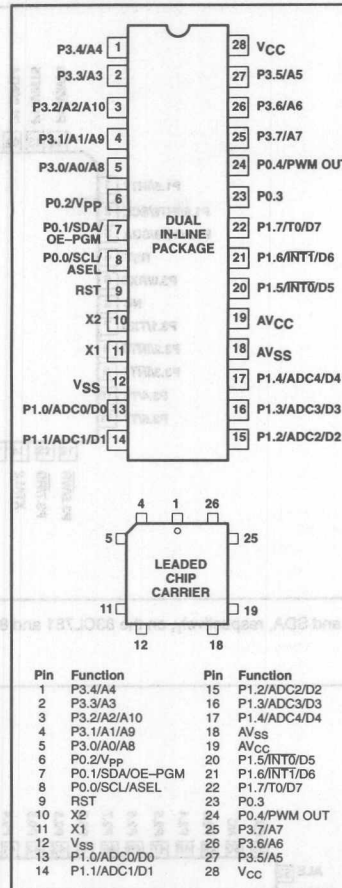
--	--	--

## 83C750/87C750, 83C751/87C751



\* P0.1 and P0.0 have the alternate functions SCL and SDA, respectively, on the 83C751/87C751.

## 83C752/87C752



## P8xCE598

